TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

THE GOOD, THE BAD AND THE UNPLEASANT – A STUDY OF GRAPHICAL USER INTERFACES IN VIDEO GAMES

Master of Science Thesis

Jyrki Turunen

ABSTRACT

Tampere University of Technology
Master's Degree Programme in Information Technology
JYRKI TURUNEN: The Good, The Bad and the unpleasant – A study of graphical user interfaces in video games
Master of Science Thesis, 94 pages, 1 appendix page
January 2017
Major: User Experience
Examiner: Thomas Olsson
Keywords: video games, gameplay, user interface design, usability, game design, graphical user interfaces

This thesis revolves around user interface design, more specifically graphical user interface design in a particular context, which in this case is video games. It describes what makes this design focus so unique and important, how it differs from "traditional" software interface design and how do you actually make a great user interface for a video game. The thesis aims to analyze existing solutions, examples and guidelines for GUI design and then provide a concrete set of instructions and tips on how any designer can improve, hopefully even drastically, their own game's user interface or at least ensure a basic, acceptable baseline for it. So far I have not come across any paper with a similar topic, so the novelty value here lies in the establishment of basic guidelines for video game UI designers regardless of their speciality.

The structure of the thesis starts with a short introduction to the topic and a description of the methodology used in the thesis. It then proceeds to talk about user interface design in general, introducing also the terms of usability and user experience (UX), graphical user interface (GUI) and game design while also providing more than one angle into the design process. In the third chapter the focus shifts onto games themselves, giving an overview of their history, what makes games unique as an entertainment method and also talking about the concepts of persuasive technology and gamification. Finally, chapters 4 and 5 focus on the video game GUIs themselves, analyzing existing examples and methods, all the while building up my own set of instructions and points to take note of when designing these interfaces. At the end, a summary section wraps up the thesis.

The methodology of the thesis relies heavily on doing research on existing solutions and using credible sources to deduce why any particular found issue was good or bad for that particular game and how it could work in other contexts. Data gathering about video games, usability and user interface design proved critical to be able to understand how to rate the value of any given solution and also enabled the creation of a list of design questions as part of the empirical part of the thesis that is then used to further scrutinize a few specific game user interfaces.

After completing the empirical part some of the findings proved quite surprising while others were solidified as the basic blocks of good UI design principles. While previously game designers relied heavily on displaying great amounts of (arbitrary) information to the player, regardless of the game genre, nowadays a trend towards minimalism can be found. Nonetheless, scrapping important information still leads to problems, so one of the mains issues for designers is to define what is important and what is not and this is sometimes a surprisingly difficult task. For this reason as well a list of questions was assembled that can help designers overcome this difficulty.

TIIVISTELMÄ

Tampereen teknillinen yliopisto
Tietotekniikan diplomi-insinöörin tutkinto-ohjelma
JYRKI TURUNEN: Hyvät, pahat ja epämiellyttävät – Tutkimus graafisista käyttöliittymistä videopeleissä
Maisterivaiheen lopputyö, 94 sivua, 1 lisämateriaalisivu
Tammikuu 2017
Pääaine: User Experience
Tarkastaja: Thomas Olsson
Avainsanat: pelit, pelaaminen, käyttöliittymäsuunnittelu, käytettävyys, pelisuunnittelu, graafiset käyttöliittymät

Diplomityössä käsitellään videopelien käyttöliittymäsuunnittelua, mikä tekee nimenomaan kyseisestä suunnittelusta erilaista muuhun ohjelmistosuunnitteluun verrattuna ja miten hyvä käyttöliittymä voidaan suurin piirtein peleissä määritellä. Fokus on nimenomaan graafisissa käyttöliittymissä, esimerkiksi erilaisia ohjaimia ei juurikaan käydä läpi. Työn tarkoituksena on analysoida olemassaolevia ratkaisuja, jonka jälkeen empiirisessä vaiheessa luodaan käytännön ohjeita ja avustavia kysymyksiä, joihin vastattuaan useimmat käyttöliittymäsuunnittelijat voivat todeta olevansa oikealla suunnalla työnsä kanssa ja varmistaa sille vähintään kohtuullisen perustason. Mikäli lukijalla ei aiheesta ole aiempaa kokemusta, työ tarjoaa perustason ymmärryksen aihealueesta teoriaosuudessa, jossa käydään läpi kaikki työn aihealueet käytettävyydestä ja käyttöliittymäsuunnittelusta peleihin ja pelien erikoispiirteisiin (vaikka tämä ei työn varsinainen tavoite olekaan). Vastaavan kaltaisia töitä en ole löytänyt, joten uutuusarvo työllä on perustason pelikäyttöliittymäsuunnittelun konkreettisen ohjeistuksen luonnissa.

Työ alkaa lyhyellä esittelyllä aihealueeseen sekä metodologian kuvauksella, minkä jälkeen se siirtyy kertomaan käyttöliittymäsuunnittelusta, käytettävyydestä ja pelisuunnittelusta tarjoten useampia näkökulmia suunnitteluprosesseihin. Seuraavaksi käsittelyyn pääsevät pelit, alkaen niiden historiikista ja jatkuen pelien ominaispiirteisiin ja mikä tekee niistä uniikin viihteenalan, sivuten samalla muutamaa muutakin muotitermiä, kuten pelillistäminen. Loput kappaleet keskittyvät empiiriseen osuuteen ja analysoivat olemassaolevia ratkaisuita, luovat ohjeistusta ja käyttävät sitten luotuja ohjenuoria muutaman spesifin pelikäyttöliittymän syvempään analyysiin. Työn lopussa on vielä yhteenveto ja työn pohdintaa.

Tärkeitä kohtia työssä on teoriaosuus, jota ilman myöhempiä kappaleita ja termistöä on vaikea ymmärtää, sekä pelien jaottelu erilaisiin kategorioihin, joissa kaikissa painotetaan hieman erilaisia asioita pelin todellisena pihvinä. Tämän lisäksi totta kai todellisen arvon luominen suunnittelukysymysten rakentamisella työn empiirisessä osassa on työn lopullinen tarkoitus, mutta sitä edeltävän pohjustusksen relevanttiutta ei voi silti väheksyä. Ilman datan keräämistä ja sitä kautta saatua osaamista ja tukea empiirinen osuus ei olisi validilla pohjalla, joten kunnon pohjatutkimuksen tekeminen oli kriittistä.

PREFACE

This turned out to be a much longer thesis than I originally envisioned. This is fine however, as it has allowed me to learn quite much about game design and user interface design in particular while working on a very interesting and relevant topic. User interface design, both in games and other contexts, technical and otherwise, is extremely important as tools and programs continue to develop to satisfty ever more complex human needs. Without great interfaces no tool is worth all too much as it requires the user to actually be able to use it, preferably enjoyably.

I've played games for some 20 years now and I'm completely enthralled with them still. Video games renew themselves year after year, classics are just as solid as in any other entertainment industry despite aging graphical quality and new innovations surfacing constantly. This makes for a very rejuvenating, self-reinventing and ever changing landscape which is incredible to follow.

If I had to do this thesis all over again, I would probably change the empirical part a bit to include actual test subjects and a variety of games with different genres and UI decisions. Now this was very much based on my own expertise and all the material I found. This decision was made because of other limitations to my work, like an actual paid employment during most of it, without which the thesis might have even been finished earlier as well. Regardless, I still found it quite interesting and a learning experience and I'm quite satisfied with the outcome.

I would like to thank all the people who helped me get through both the writing of this thesis and all the studies leading up to it. So a big thanks to all my friends and family! I would especially like to thank my thesis supervisor Thomas Olsson who provided me excellent guidance and feedback whenever I required it.

If anyone has managed to read through the thesis, hopefully you found it enlightening and entertaining as well. It could easily be expanded to include actual user testing, not to mention reaching out to game companies and gaining more information and insights from them from actual projects and experiences. The possibilities are vast and the topic is still very up-to-date as video game development is a big part of modern IT world. However, for now, I'm looking to actually develop some games myself and see how far I can get with that. Thank you for reading and game on!

TABLE OF CONTENTS

ABBREVIATIONS & TERMS

Term	Explanation
UI, user interface	The interface with which the user interacts with software/hardware, for example a keyboard or the front page of a website.
UX, user experience	The subjective experience of using a product/tool/software.
GUI, graphical user interface	Like a user interface, but in a visual, graphically presented format, usually on a screen or monitor of some kind.
RPG, role-playing game	Video game genre
Esports	Electronic sports, or Esports, is the sports industry of playing video games professionally.
FPS, first-person shooter	Video game genre
HCI, human-computer interaction	Term for the process where a human performs actions with a computer and the machine reacts in a desirable way, accomplishing some manner of a task.
HUD, heads-up display	HUD is a collection of information displayed on the screen of a game that indicate player status (health bar, ammo counter, navigational aides etc.)
Franchise	A known title that spans multiple different games of the same type, for example *Assassin's Creed* or *Final Fantasy*

.

1. INTRODUCTION

Games, as an entertainment or even an education method, are only as successful as their method of conveying the elusive “fun factor” to their consumers, the players, via a user interface. Generally, this interface is a graphical one that is viewed on a screen of some sort, although other interfaces do exist. For the purposes of this thesis, I will be focusing on graphical user interfaces in video games as reviewing each type of an interface would merit a thesis in their own right. After an introduction to games and their variety of user interfaces, I will focus my efforts into creating a solid set of guidelines for video game user interface design as a collection of such could greatly increase the ability of newer, more unexperienced designers at the early stages of their careers.

Video games have, especially over the course of the last decade, solidified their place as one of the main venues and markets for entertainment alongside television, movies, music and other similar, more traditional or long-standing methods of fun and enjoyment. With the bigger games' budgets counted in the millions, sales counted even higher, the over-abundance of bite-sized games on Facebook and mobile devices and the emergence of Esports, games have permeated the everyday life of almost everyone, from a small child playing Candy Crush on an iPad to an elderly person playing Wii Sports with their grandchildren. Video games have come a long way in the four or so decades after the first game of Pong in 1972 and show no sign of stopping the advance. This makes understanding games and what makes them so much fun a pivotal point for anyone wanting to understand the video game market or even human behavior when it comes to seeking and enjoying entertainment. [1]

So, what separates games from other similar entertainment industries, like movies? Interactivity. In games, the player is in the spotlight. Even though most games have a main character or an avatar that the player controls and who moves and interacts with things inside the game's world, nothing would happen without input from the player. The player pulls the strings and accomplishes things in the context of the particular game he/she is playing while simultaneously, hopefully, having fun. This is where game design, and user interface (or UI) design, becomes a critical factor. If what the player sees on their screen is somehow illogical, wrong, in the way or continuously breaks player immersion in the game, the fun factor is decreased and the player won't enjoy the game, up to a point where they will stop playing it altogether and possibly never even try again. [2]

When it comes to video game graphical user interfaces (or GUIs), the same rule applies as when meeting new people: you only get one chance of making a first impression and the better it is, the more likely the other person/player is to interact with you in a desirable way. For video game companies, this is literally a make or break-situation: if your game's UI is deemed horrible and drives customers away, these same customers won't most likely give your other products much of a chance as alternatives in every genre are aplenty. Naturally there are exceptions and bigger companies have entire franchises where one bad game won't sink the boat, but it won't boost sales either. Also, when it comes to franchises, the basic UI is very rarely something terrible to behold as it has been honed to excellence over the course of the previous installments in the series. Nonetheless, for new companies or new brands, this is a very serious challenge. Game design as a whole can be one big minefield without the proper guidance and some sort of understanding of what makes games fun and user interface design is only a part of it, but it's importance still cannot be waved off. [3, 4, 5, 6]

Naturally, a plethora of UI design guides and traditions exist so anyone wishing to create a game themselves doesn't have to start from scratch. Many decisions regarding UI design can also be done utilizing a little common sense, for example you don't put the ammo counter in the center of the screen with big colorful letters in a first-person-shooter (or FPS). Somewhat illogically you can, however, fill the screen with colors and colourful elements if they are there to serve a purpose or enchance the gameplay experience as for example *Destiny* did and it proved to be a huge success. For those designers who do not want to make yet another copy of the genre's most successful release or who want to try something new or different, the road to success can be perilous. Pushing the boundaries is generally all well and good, but when does a unique design solution simply translate to bad design? Where's the fine line between creating something ground-breaking and something that makes players, or users in general in other contexts as well, cringe? This is usually not clear, especially as some design solutions only become terrible/great over an extended period of time. [7, 8]

1.1 Methodology

The methodology used in this thesis is based largely on literature reviews and experts analyses of game designs and relying on material found about the topic from credible sources. There were some restrictions time- and resourcewise when this thesis was being created, which led to the decision of not including outside test subjects or user testing, for example. The thesis is more of a study on what really makes a successful

video game user interface, using both the writer's own extensive experience in the field as well as solid measuring tools gathered from outside, independent sources, for example the heuristic analysis methods of the renowned usability guru Nielsen.

The material was gathered using specific keywords to begin the search, such as “video game user interface design” and “user interface design best practices”. A vast amount of research and work has already been done in terms of user interface research, so that was a logical place to begin. Afterwards I expanded into video games, looking for examples of all kinds of design solutions in them and comparing them to the reception the games received in the eyes of the public, critics and experts alike. Following the links in the material I found myself I quite quickly gathered a reasonable amount of source material to begin my work. Analysis used both my own long experience in the field and various outside tools, such as the tested and tried heuristics from Nielsen, in quality measurement.

Building the understanding piece by piece throughout the thesis, conclusions and deductions are very logical and natural and the overall outcome seems like it was simply there, waiting to be discovered. Nearly all of the material for the thesis has been gathered from the web as it is immensely easier to find a wider pool of data from there without sacrificing credibility or having to browse through massive libraries filled with irrelevant material.

1.2 Goals and structure

I have always been a big fan of great user experiences in digital formats, which naturally made me a fan of video games as well. For a long time now I've wanted to find out what exactly is it that makes for great games and breaking down their user interfaces is one part of that effort. UI is the most visible part of any game, so it is a great, and somewhat under-researched, starting point. Therefore, ultimately this thesis aims to provide a concrete set of guidelines for video game user interface design that anyone can use to boost the quality of their interface. The main questions I aim to answer is how can you make a good video game user interface and what do you need to consider in it?

In this document I will first (in chapter 2) go through what user interfaces are, what UI design is, how is user experience (or UX) related to UIs, give some examples on a variety of successful and not so successful GUIs both in video games and elsewhere and introduce some traditional UI design patterns and guidelines. In chapter 3 I will then dive into the world of video games with a brief historic on them, after which I will review what games are and what types of games there are, how are game UIs different from other UIs, review some game design traditions with a focus on UIs,

introduce a concept called “gamification” and shortly discuss about persuasive technology. In chapter 4 I will go through both good and bad examples from a variety of game genres and try to glean similarities, differences and lessons to be learned from them as well as provide a list of questions which can help video game designers when creating their own interfaces. The chapter is about breaking down game user interfaces to learn what makes a successful game UI and what does not and how can these lessons be utilized. On that note, in chapter 5 I will be collecting everything I've learned and come to notice based on the previous chapters and providing concrete notions on what you should think about when creating each particular type of video game user interface view. The concept design will thoroughly list and weigh different options for each part of the UI and in what contexts might any particular solution be the most appropriate. Finally, in the last chapter I will give a summary about the thesis, discuss its strengths and weaknesses and conclude the document with some words about possible future work on the subject.

2. USER INTERFACE DESIGN

Imagine the most beautiful scenery you can think of. It can be a glorious sunset upon a windswept hill with the ocean lapping gently against the stones far below, the face of your loved one smiling at you on a sunny day or whatever you wish it to be. Now, what is it exactly that enables you to enjoy and take in everything in that particular view? Your eyes, surely, so you can see it. Perhaps also your ears so you can hear the sound of the wind, your nose so you can smell the salt of the sea and your hands so you can touch your dear person. Good, now think about the following: technically speaking, you've just utilized a variety of human user interfaces with which everyone interacts with the world around us. All of them have developed over untold thousands of years to provide us with the capabilities we have today and where they haven't exactly been as much intelligently designed as they have evolved to fulfill a variety of necessities, that can be viewed as genetic and contextual user interface design.

So, basically, all of us are familiar with the concept of user interfaces even if many do not know the term itself or the text-book definition for it or even realize it. We expect our senses to provide venues for us to live our daily lives and interact with the world around us and have developed many businesses on the grounds of providing aid to these functionalities if they are somehow “broken”: eye glasses, hearing aids and wheelchairs are just a few of the most common examples of these.

Now let's look at the other end of the interaction. There's a variety of objects and things that we interact with and react to on a daily basis, be it light switches, foodstuffs, rainstorms or a million other possibilities. More often than not, most of these interactions have been repeated so many times that they've become automatic, but just as many things out there require conscious effort and focus to do. Nowhere is this more prominent than in the workplace. Regardless of if you are a bus driver, a professional athlete or a software developer, what you do for a living makes you concentrate and aim for results that require you to actively think, act and conform to new situations.

In the modern world, almost everything is computer-driven and has some sort of computing system implemented within it. Naturally this is not the case for some physical objects, like footballs and books, but even they are boosted to a new level when technology is involved. Football matches can be televised for millions around the world and books can be written, sold and read anywhere without the need for an actual physical copy of it. Technology has permeated every level of human life in some way

and technology use has become just an essential a skill as seeing and breathing for most people.

However, as this dependency is artificially generated, it's not as automatic as the human body's natural functions. Most devices require some kind of instruction and learning to use and in many cases the use of the device might not be intuitive from just the look of it. To make technology as user friendly and easy to use as possible, we need effort and planning. This is where user interface design comes in.

2.1 Defining UX and UI

User experience, shortened to UX and user interface, shortened to UI, are similar concepts and closely related to one another, but they are not synonyms for one another. Many times these terms are used interchangeably, but they shouldn't be, as they refer to different things and have different meanings. [9]

User interface refers to the methods and interfaces with which users interact with, well, basically anything, be it a graphic user interface on your screen while you play a game (with the screen technically being another interface as well), a remote control you use to change the channel on a television or even a simple fork you use to eat. User interface in its most simplistic form can be viewed as a tool to enable the user to accomplish a goal. Granted, UI as a term is not perhaps widely used outside of the IT industry, but the definition can still be applied to many things. Most commonly user interface refers to the elements on a screen that show the user information and enable them to interact with that information, be it creating a PowerPoint slide or hacking off the head of an attacking orc in a video game. [9]

User experience, on the other hand, refers to the somewhat vague experience of using said interface. UX measures how intuitive, enjoyable, efficient and easy to use the interface is and UX design aims to increase the overall usability of the interface through established patterns, design solutions and decisions. User experience is not something physical and concrete, but it is irrevocably a part of all UIs. Successful UX design may even in most cases go completely unnoticed as the tool or interface you are using is efficient enough in doing its job. Only in cases where the design is exemplary or abysmal does it usually pop up. For example, in games, most of the complaints in the forums are related to user experience (this hero in this game is overpowered, I can't see this thing on the screen, feature Z is missing from the game).

To put it simply, Desi Quintas offers an excellent example of the difference between UX and UI:

"UI, or User Interface, refers to the methods (keyboard control, mouse control) and interfaces (inventory screen, map screen) through which a user interacts with your game. UX, or User Experience, refers to how intuitive and enjoyable those interactions

are.

To look at it another way: the UI of a car is its steering wheel, its pedals, the dials and controls on the dashboard; the UX of the car comes from intangibles like the brake pedal being responsive, the engine smoothly accelerating when you step on the gas, the gear stick having just the right amount of resistance - those things that make the car enjoyable to drive." [9]

In the context of this thesis, UI is attributed to mean video game user interfaces, more specifically graphical ones that the player sees on their screen. I will not be discussing physical user interfaces like controllers or output devices apart from a few occasional mentions. This paper is focused on graphical user interface, or GUI, design.

One last term that should be mentioned in the same breath as UI and UX would be Human Computer Interaction (HCI). HCI at its core is the study of interactions and information flow between a human and software through an interface. Usability is a major focus point of HCI and overall HCI wants to find out what makes the most effective, efficient, safe, learnable, memorable and utilizable interaction possible. The discovery of what makes a “good” user interface and a “good” user experience is the domain of HCI. Especially when interfaces and software become more and more complex, HCI studies step up in importance as a means to control and understand the complexity in an effective way. Seeing as video games are a form of software, the inclusion of HCI in both design and evaluation processes is quite natural and can provide valuable insights into creating better quality games. HCI fits in nicely with the concepts of UI and UX design as a sort of mediator between the two, including many aspects of both and yet providing its own insights into the processes required in software development. [10]

2.2 UI design vs. GUI design

Designing user interfaces comes in as many a form as there are interfaces in the world. Though some usability principles may be the same when designing an inventory screen for a role-playing game (RPG) or trying to organize all the widgets and levers in the cockpit of an airplane, the end goals and critical aspects for both differ wildly. If you miss a click or mistakenly push something while browsing your inventory, it's not a big deal. If you push something you're not supposed to while piloting a plane, the consequences can be... not ideal. As video game design does not affect anyone's physical well-being or other important functions, but is simply a form of entertainment to be utilized when desired, video game designers have a lot more freedom when it comes to their design solutions. They are free to play around with different ideas whereas cockpit designers are not. Indeed, similar video games in the same genre use

their visual appearance as one main way to separate themselves from their competition. Their graphical outlook is not only one of their main draws in the customer market, it's also a defining trait that especially franchises utilize to a great extent.

However, in the context of this paper, the focus point is on graphical user interfaces viewed on a screen, most often than not a computer monitor. Even so, there are still distinctions to be made between “traditional” UI design and GUI design. By UI design I mean interfaces built for utility applications, workplace software and web browsers. GUI design then means video game user interface design, where the UI is there to be part of the immersive user experience in addition to being the main tool to influence events in the game with.

Most notably, when you're designing a GUI, you have complete control over every pixel on the screen. Video games are designed for specific platforms, be they PCs, consoles or mobile devices, but the uniting fact is that you KNOW what devices you are designing for. This enables you to learn their limitations even before the design has started and conform your design decisions accordingly where needed. Whenever you see a game interface, you can be sure that the developers made it look more or less exactly like they wanted to. [11]

At the other end of the predictability scale, we have non-game software and, especially, web design. When you're designing an interface for browser use, all assumptions fly out the window. I've come across many a developer, and struggled with it myself as well, that felt like ripping some hairs out of their heads while trying to modify their application to work on most browsers and their older versions that don't necessarily support the same functionalities that, in modern day web design, are the norm. This is especially a problem in a variety of workplaces that have not updated their hardware and/or software for a long time and are still working with browsers like Internet Explorer 7 or 8. To add insult to injury, pages on the web can be viewed on more or less any kind of device from a computer to a smartwatch. So not only do browsers have very different requirements, the devices they're run on have them as well. Add in the problems of stable internet connections and viewing pages on the move and you have quite the complicated concoction of variables that simply have to be addressed when designing your interface. [11]

Basically, it is impossible to design for web and design for games using the same mindset as the contexts are so different. Other software, like work applications and utility applications, like Microsoft Word etc., fall in between these two end points. Much of what can be used to design games can be utilized in designing their interfaces, yet the end goals are still different: where games exist to entertain the player, most other applications exists to do so as well, but to provide us the capabilities to do a variety of things, be it listening to music, drawing beautiful art pieces or just simply creating a PowerPoint slideshow to present at the next work meeting. Where game interfaces can

be stylized and convoluted a bit to make room for “the rule of cool”, meaning that if it looks/feels/seems awesome it can stay, other software tend to value efficiency and ease-of-use over entertainment.

A brief list of UI design principles has been compiled by Janko Jovanovic (2009) that, although old, can still further shed light on the different solutions available. They revolve around what is displayed to the user, what actions they can take and in what points of use and overall the look and feel of a user interface. These design patterns are mostly for browser based applications and web pages (like lazy registration, which means that the user can take actions and try out the website/product before registering, and account creation, where some content requires logging in and it becomes burdensome if too much information is required too often), but some of them can also be applied in other software contexts as well (like progressive disclosure and hover controls, discussed in the empirical part of this thesis). This just goes to further emphasize both the differences and similarities in traditional UI design and GUI design as neither have a single fixed method of creation, but instead can be as varied and unique as the designers and developers themselves. Innovation requires breaking barriers, and user interface design is no exception. [12]

2.3 Game design

Game design as a term can be understood as encompassing more than just video games, for example board games and social party games. Indeed, the underlying principles behind all games are to entertain and to let the players have fun, either on their own or, more often than not, with other people. As games in general have much in common, a (video) game designer can draw from a vast pool of resources and experiences when trying to find a solution to a specific issue or problem or design a game mechanic to be as enjoyable as possible.

However, narrowing the focus to the topic of this thesis and video game design in general brings some limitations and possibilities unique to HCI solutions. Whereas board games normally require you to gather your friends in a single physical location on an appointed time and date, video games suffer from no such restrictions. Internet-based games and communication channels are a given in modern video games. Your friends can be on the other side of the globe and you can still mow through hordes of zombies together while discussing the day’s events. Then again, video games don’t usually allow for improvisation, “ad-hoccing” or house-ruling to the same degree as other types of games, but enforce their built-in limitations and rulesets to the letter.

The bigger difference can be seen when comparing video games to utility applications like Office tools or anti-virus software. These applications are not made for the main purpose of entertaining the user, but to accomplish a certain goal in the most

effective way. Games are meant to be fun, but you don't want your computer's firewall to focus on being flashy to use instead of being a solid protection against malicious software that might attempt to invade your computer. Where utility software is designed to be as efficient, easy to use and streamlined as possible, games can bend these rules in order to provide a more in-depth gaming experience (naturally these attributes should not be forgotten entirely). [13]

A good comparison is the application of cognitive load on the user: utility applications aim to minimize the mental capacity required to perform most tasks. Games shoot for the opposite; they are meant to be challenging and require the player to actively take part in them and perform actions to influence events in the game, produce desirable outcomes and, therefore, pleasure, meaningful play and feeling of enjoyment. Naturally some of the end-goals are the same for some utility applications as well, but how many people do you know who use PowerPoint for fun? In some aspects, games can be more likened to amusement park rides with a touch of scriptwriting and storytelling than other software applications. Where utility applications strive to minimize challenge, games seek to control and adjust it but always keep it on a noticeable level. [14, 15]

Nonetheless, as both video games and utility applications ARE software at the end of the day, they have plenty in common too and can benefit greatly from one another. Lessons derived from game design can be helpful for utility applications (see "*2.5 Gamification*") and the pursuit for efficiency, especially in terms of user interfaces, can provide game designers valuable examples and solutions to problems they may themselves encounter. [14]

2.3.1 Ludology and the designer's dilemma

Ludology, which is the study of video games and most related phenomena, offers a concept where games are broken down into two categories based on the type of play involved: *paidia* and *ludus*. Of the two, paidia is much more freeform with fewer rules and restrictions where ludus is considered the "serious gaming" half, where there are strict rules and by the end of the game the participants are separated into winners and losers. Paidia can be understood as children's play, where rules are made up (if not completely arbitrary) where freedom of thought and having fun is more important than strictly adhering to pre-set restrictions. Great examples of ludus are traditional games and sports, like soccer, Monopoly or simple tag. Both of these terms can help video game designers understand more about their would-be game if they simply identify in which category their game belongs to. For example, *Minecraft* (2011) is clearly an example of a paidia-type of game, whereas *Minesweeper* definitely identifies as a ludus-type of game. Naturally, simply belonging to one category or the other doesn't mean there is no flexibility or room to improvise, but as game designers have a plethora of

difficult decisions to make during the development process of their game, even a simple identification like this can make a world of difference when it comes to focusing their vision on the game. [16]

Video game designer also face problems that are not present in any other field, at least to the same degree. Many developers continue to describe a career in video games "a calling" as it's not the most financially rewarding or stable of industries nor is it one where things happen easily or without compromise. A lot of great ideas never come to fruition because either time, money, the required skills or motivation (or any combination of the four) are missing/inadequate. Game designers too often have to abandon some of their lofty ideas for their brainchild as it is simply not possible to achieve all of them within the space of a limited time and budget. Falling in love with your idea is a risk of the biggest kind, as more often than not the end result is at least somewhat different than the original idea. This of course depends on a lot of variables during development and the risks grow as the scale of the game grows, but it is something that is ever present within the games industry. [14]

Another issue with video games is the target demographic or the target group of a game. Unlike in other applications, the audience that plays a particular video game can come from basically any background, culture, ethnicity, age, gender or moral environment. To accommodate everyone's specific needs is impossible and game designers tend to walk on thin ice whenever they make unusual decisions regarding their game's characters, world or mechanics. Someone, somewhere will more than likely get offended by something in your game. This is something you just have to accept and strive to justify your means with your goals. Games are much like music, in a sense: no single game can satisfy the needs of everyone nor are they supposed to. Where non-game application developers have to carefully work based on their expected user group's preferences and desires, game developers tend to be a bit more unbound by that particular restraint as predicting who is going to play your game is like predicting the weather: sure you can get a general idea about it, but it all can change in an instant and you will never be a hundred percent right. The designers and developers themselves might not even be part of the target group, and so must seek insights about the desired gameplay choices elsewhere instead of only relying on their own experiences as players. [14]

2.3.2 The challenge of challenge

So where does the challenge come from in games and how do you control it? Games often tend to have difficulty levels where the player can choose how tough the enemies are or how crafty the opposing AI (artificial intelligence) is. This of course is not the case in every genre, as for example in casual games the challenge is usually pre-

designed into the levels or just randomly generated. The common ground for all games and their challenge sources, however, is that they all are related to the gameplay and the mechanics themselves, not the user interface. There are exceptions that make the UI part of the game and therefore part of the challenge, but in general the UI is there to provide the player the avenue of interaction to overcome the challenges in the game itself. Designers create challenge by manipulating objects in the game's world and setting rules as to how they interact with and affect each other. A good example of this is the rock-paper-scissors-type of setting of many strategy games: a unit of spearmen will stop a cavalry charge in its tracks, but will struggle against archers, who in turn will get run over by cavalry. Everything is situationally good and it is up to the player to maximize the effectiveness of the tools at their disposal while trying to prevent the enemy from doing the same. [15]

In some cases the challenge from a pre-set bunch of rules can be learned and mastered which could make a game boring as the challenge has been overcome and the player can always make the best possible decision and defeat any obstacle thrown their way by the game. This is where random events help to balance the scales. Games and game designers treasure predictability just as much as unpredictability when it comes to designing gameplay challenge. Random events are out of the player's control and make it possible for basically anything to happen at any time. Keeping to the example of the strategy game, imagine you have the most well-defended castle in the entire realm with thick, tall walls and a myriad of soldiers to stand guard on them. Then, out of nowhere, a plague breaks out inside the keep. Now your stronghold becomes a trap for all inside and you as the player have to make some critical decisions that came upon you by sheer chance. Or maybe a contingent of the soldiers rebelled and opened the gates for an attacking force? A certain amount of random events and possibilities can keep a game from becoming monotonous and predictable, which in turn helps the player enjoy it more and experience the game as challenging for a longer time. This is yet another difference between games and utility applications, as the element of randomness is something that is systematically eliminated from non-game and non-gamified software. [15]

However, there is more that game designers have to take into account when making their games. Randomness can be a valuable tool in creating diversity within a game, but it can also be a deterrent, depending on the type of game. In narrative-based games randomness is usually controlled to some degree and a lot of the game world is pre-designed, meaning that despite seeming vast, the game world won't change in subsequent playthroughs. Narrative-based games rely on certain things and people being in certain places and key events taking place when a set of conditions have been met. The story can't rely on the variable of randomness. Then again, absolutely predictable enemies and situations will lead to boredom, so during non-critical moments in the game randomness can still be valuable, for example when spawning enemies on the

map. Victory is mostly due to the player's performance in doing the correct things and following the path laid by the designers, but also partly due to reacting to unpredictable situations and scenarios, yet still coming out on top. [16]

At the other end of the spectrum are simulation-based games. While the simulation part should be as accurate, learnable and repeatable as possible, it is the random events and situations that provide the challenge for much of the game after the controls have been learned. Designers make challenges and achievements for such games, but a lot of the fun can be extracted from perfectly controlling your plane/truck/train/whatever you're simulating in unforeseen situations. The very idea is that you never encounter the exact same situation twice and have to adapt to new events using the same rules and tools. Simulation games are also unique in the aspect that despite the end result might look the same to the casual observer (you're still playing a game), the means by which the gameplay enjoyment is achieved differs vastly from many other types of games. Simulations are about doing the things yourself and mastering the controls of the simulation, not just playing a game. A good comparison is the playing and watching of football as the same principles roughly apply. [16]

Finally, another big factor in the design of a game's user interface, somewhat paradoxically as the user group can be defined only approximately, are the players that will eventually play the game. How this is actually done varies from developer to developer and game genre to game genre, but regardless of your specific chosen method, involving the players in the design process can provide key insights as to how the game should be played, what parts are good and what needs to be worked on. In some cases, like massive online multiplayer games (for example *World of Warcraft*, 2004), the players input can be directly seen as UI extensions, add-ons and mods, that can be freely developed and distributed. Some can even be included in the game's own UI if they are popular enough. In other cases alpha or beta testing can be a way to get feedback from the players before the release of the game. There exist a wide variety of ways for collecting player feedback on both the UI and the overall gameplay of a game before it is completed and bringing in this kind of fresh and unbiased views can sometimes prove critical to the game's success. [10, 14]

Considering how games hope to “bring about a state of affairs using the less efficient means possible”, usability in video game design can be a fickle thing. You have to provide the player clear goals and the means to achieve them in a fun way, but you also need to enforce the rules and restraints set by the game and see that the player still enjoys the experience. A good example of the quote above is golf. The objective in golf is to complete the course by successfully getting the ball into all eighteen holes in the course. Naturally, the most effective way to do this would be to carry the ball into all of the holes, but the rules of golf prevent you from moving the ball with anything other than by the means of a golf club, giving the player a less efficient method of achieving the goal. The usability of the golf club therefore becomes a priority as it is the main tool

with which the player willingly pursues her goal. The same applies to video games but NOT to utility software. [15]

2.4 Usability versus business goals

Usability is an important part of any user interface, be it virtual or physical. The very nature of software and video games is that they are artificial and have been created by people to complete a certain task. This can make software in general difficult to use as not everyone thinks alike or would go about finishing said task in a similar way. When designing user interfaces, I've found that a simple rule of thumb usually helps a lot when trying to solve issues with you design: people are dumb, don't assume they know how anything works. Usability as a professional field aims to make UIs as easy to understand and effective at their job as possible, but regardless of how much work you put in and how big and flashy your guiding arrows and instructions are, somebody will fail at using your interface. Somewhat ironically, user interface is nonetheless one of the most important aspects of any game and the simple fear of commercial failure tends to be enough for most developers to hone it through countless iterations and keep improving it even after releasing the game. [10]

The question then becomes, when is your game's UI good enough? If somebody will fail to grasp its usage no matter how many design iterations you go through, when do you conclude that it is ready for release? The answer more often than not comes from a somewhat unexpected quarter: the UI is ready when the release schedule, budget and time restraints dictate it is ready. All of us have probably at one point or another created something, be it at school, at work or on your free time, that you wished you could go back on and improve as soon as you published it. Such is the case many times in video game UI development. The minute details could be switched around for all eternity, the color scheme of a particular inventory piece is ever so slightly off or the notion of being hit by an enemy doesn't show up exactly the way you want it to, even though it was perfect the day before. Usability is not an exact science, but the best estimate of one or more people who have some understanding of what it means and how it can be applied to a specific situation as well as it can be to enable as many users to use the product as possible.

Video games are development projects with more or less fixed budgets and schedules. You can't expect to be honing your UI forever or have infinite resources available for it either. A lot goes into video game development and only a part of it involves UI design. Sometimes the schedule simply cannot budge and your UI is still not ready in your estimate. Sometimes the coding of the game takes longer than it should have or some of the resources allocated to you are repurposed. There's a wide variety of reasons why the UX and the usability of your UI can be compromised and there might not be anything you can do about it. Sometimes usability can be compromised in the developers simply not understanding what they're making at all, as

can be seen from the disaster that was the first launch of *Final Fantasy XIV* back at 2010. In an ideal world, you'd have plenty of time to hone the UX of your game to its absolutely peak, but in the real world you have to make compromises and do what you can with the resources available to you. [17]

3. GAMES

Games, in general, have been around since we could walk, even before we could talk. I'm sure stone age humans enjoyed some form of tag as children and/or other similar games in between grunting at each other and their parents. You could argue that playing games is part of our heritage and a big part of everyone's life, it's just that the types of games we play can change over time and age.

More recently, the emergence of video games has transformed the way we play and enjoy gaming as a whole. Naturally, there will always be a place for school-yard games, board games and the like even if video games have become the big player in the field. Just some short 40 years ago video games were only an infant field, with games like Pong and Space Invaders being the most visible pioneers for a whole new entertainment method that would go on to revolutionize the gaming industry and many people's daily lives as well. Electronic sports, or Esports, are gaining ever more popularity and the video game industry rivals or even surpasses any of the more traditional entertainment industries like movies or music in yearly net-worth. [1]

Considering how well the industry is doing for itself, I'm going to next provide a brief overview of how it all started and how games have evolved over their four-or-so decade long journey. In this chapter I will also divide games into specific genres I'll be using later on as well as touch on the topics of persuasive technology, UI elements in games and gamification.

3.1 A brief overview of games and their history

Before delving into the history of video games, let’s take a quick step back and ask ourselves, what constitutes a game? What is it that makes us define a certain entertainment method or a whole as a game? Ludology, or game studies, can provide us a definition or two that I find can answer that question. [16]

In game studies, games are seen as a collection of multiple necessary conditions. Where individually they are just rules and behavioral concepts, a combination of them can constitute a game. A bit older definition for a game is “exercise of voluntary control systems in which there is an opposition between forces, confined by a procedure and rules in order to produce a disequilibrial outcome”. This describes a video game pretty well, but more recent definitions exist as well. One that fits the bill pretty well reads as follows: “a game is a rule-based formal system with a variable and quantifiable

outcome, where different outcomes are assigned different values, the player exerts effort in order to influence the outcome, the player feels attached to the outcome, and the consequences of the activity are optional and negotiable." Optional and negotiable consequences can be understood as the outcomes of the player's decisions and actions when she is presented with more than one plausible course of action. Playing a game always empowers the player to influence the outcome of the game within its specific set of rules and restrictions and the outcome of the game is the result of the player's decisions and efforts countered by actions from other players or the game system. Whoever utilizes the set tools and methods allowed by the game in the most optimal way generally comes out as the winner or the most successful player. [18]

Armed with this knowledge and keeping the definition in mind, it's time to look back on the history of video games.

1950-1970

The dawn of video games. While game-like concepts have been created before, like programming a "primitive" computer to play chess or Tic-tac-toe, the first "official" video game is developed in 1958 by physicist Willy Higinbotham. The game simulates table-tennis and can be played on an oscilloscope. Four years later, in 1962, the first interactive video game is created by Steve Russell. The game is called *Spacewar*, it runs on a Digital PDP-1 mainframe computer and the graphics are ASCII text characters. Over the next ten years, it spreads to computers across the USA. The programming language BASIC is created in 1964 by John Kemeny and as a result programming and video game creation starts spreading. In the latter half of 1960s Ralph Baer comes up with the concept of playing interactive games on a television screen and starts to explore the idea further. [19, 20]

While the video games of this era are still in their infantile stages (which can clearly be seen from simply the amount of events in this period in comparison to later decades), the foundations are laid and the idea of interactive games is born. The motivation and the innovation to produce video games is there, but the limitations of technology restrict developers to very crude and simplistic designs. Nonetheless, the pioneers of the industry start emerging and video games as a concept starts developing.

1970-1980

Nolan Bushnell and Ted Dabney, the future founders of Atari, develop and publish the first video arcade game called *Computer Space*, which is their version of Russell's *Spacewar*. Baer patents "A television gaming apparatus and method" in 1972. Having

licensed Baer's TV game in 1970, Magnavox releases *Odyssey* in 1972, the first-ever home video game system. That same year, Atari releases one of the most recognizable games ever, *Pong*. *Maze Wars*, a predecessor to every first-person-shooter ever, is released in 1974. Home version of *Pong* is released in 1975, bringing video games to people's homes instead of arcades. The first console which features the ability to change games by changing cartridges is the Channel F released in 1976 by Fairfield Camera & Instrument. In the same year, the grandfather of modern role-playing games, or RPGs, the text-based game *Adventure*, gains popularity. Following in Channel F's footsteps, Atari publishes its own cartridge-based console Atari 2600 in 1977. Some of its biggest features are games in color and adjustable difficulty levels. In 1978, Midway releases *Space Invaders*, which becomes an instant hit and actually causes a shortage of 100-yen coins in Japan. Mattel develops a competitor to the Atari 2600 and releases Intellivision in 1979, which features better controls and more sophisticated graphics. That same year, *Asteroids* becomes the first game to allow players to store three-letter initials on a scoreboard in the machine. Finally, 1980 saw the birth of *Pac-man*, which became the first video game to be ported from arcades to home consoles (Atari 2600). Also in 1980 the first games incorporating 3-D and a virtual world were released, with *Battlezone* including the former and *Defender* the latter. [19, 20]

It is a decade of firsts as advancements in technology allow game developers, perhaps not yet labeling themselves as such, to bring their ideas to life in more complex ways. This decade foreshadows greater things to come and gives us a lot of legendary games and game figures like *Space Invaders* and *Pac-man*. Video game consoles also make a beachhead into people's homes with the Atari 2600 and Intellivision and generally start spreading the word of video games to a much wider audience. Gaining momentum and seeing the business opportunity that the video game industry can provide, companies like Atari, Activision and Namco actively seek to develop and release better and better games and consoles.

1980-1990

The first video game magazine, Electronic Games, is founded in 1981 by Arnie Katz and Bill Kunkel. That year also gives birth to perhaps the single-most recognizable figure in video games as Mario makes his first appearance in *Donkey Kong* (named as Jumpman in the game). 1982 saw two more consoles, the Atari 5200 and Colecovision, released and competing for customers. The first video game related movie *Tron* is released by Disney alongside a game with the same name. *Ms. Pac-man* strikes a blow for gender equality by becoming the best-selling arcade game of all time. A year later, in 1983, the first game featuring laser-disc technology, *Dragon's Lair,* is released. The

most powerful video game console to date, Commodore 64, sees the light of day. In Japan, the 8-bit Famicom, later known as Nintendo Entertainment System a.k.a. NES, is released. Though it takes two years for the NES to make its way to the United States, it almost single-handedly revives the ailing video game industry there. One of the most legendary games of all time, *Tetris*, is created in 1984 in Russia by Alexey Pajitnov. To compete with the NES, both Sega and Atari release new consoles, the Sega Master System and the Atari 7800 respectively, in 1986. In 1987 RPG fans everywhere rejoice as the future game development legend Shigeru Miyamoto creates *The Legend of Zelda*, the popular pen & paper RPG *Dungeons & Dragons* gets licensed for video games and the first of the infamous *Leisure Suit Larry* game series brings a whole new kind of maturity to video games. The next year it's time for sports fans to get their turn in the sun as *John Madden Football* hits the game markets. While hand-held consoles had been around in the past, it was Nintendo's Gameboy that really broke the scene wide open with its release in 1989. Atari tries to compete with the Gameboy by releasing Lynx, but despite good reviews and decent market success, it never quite matches up to Nintendo's console in sales figures. On the console front, the first console using compact discs, the TurboGrafx-16 makes an appearance and the first ever 16-bit game console, the Genesis, is released by Sega. Video games get one of their biggest breakthroughs in 1990 when Microsoft includes a video game version of the popular solitaire card game in Windows 3.0. People who otherwise wouldn't play became enthralled by the boredom/time killer and millions enjoy video games for possibly the first time. [19, 20]

The 1980s is truly the decade of consoles. The rivalry between console publishers rages fiercely and various competitors enter and exit the scene during this time. In hindsight, the success of the NES and later on the SNES was definitely not a given, but the ease of use eventually won the hearts of the masses. Atari started struggling and while Sega did ok, it never enjoyed the same kind of popularity as Nintendo's consoles. The modernization of hardware is starting to show, with games no longer hard-coded or included with the consoles but bought separately on cartridges or discs. The decade also included the genesis of Mario, Tetris and The Legend of Zelda, which continue to hold great deference within the video game industry even today. Even the themes of video games really evolved during this time period, starting with the simple barrel dodging of *Donkey Kong* and going all the way to cinematic story-telling in *Dragon's Lair*, exploring an open world with recognizable and personable characters in *The Legend of Zelda* and even including adult references and content in *Leisure Suit Larry*. Games were no longer simplistic in both execution and nature, but started to require and offer more and more ways to interact and deeper game worlds that began to hold emotional meaning for the player. Somewhat ironically, considering the previous

points, the gamifying of casual solitaire is what made video games known and played with an incredible number of people who otherwise might not even register them at all.

1990-2000

Super Nintendo (or SNES), the 16-bit successor to the wildly popular NES, is released in 1991 (a year earlier in Japan). Sega looks for a trademark character to spearhead its console sales and Sonic the Hedgehog is born. *Dune II* puts real-time strategy games (or RTS games) on the map in 1992, while *Doom* does the same for first-person shooters (FPSs) in 1993. The video game rating system is created as a result of the extra graphic violence portrayed in *Mortal Kombat*. A year later, in 1994, Sega Saturn and Sony Playstation are released as well as the first installment of the future hit game series, *Warcraft*. 1995 brings about yet another console release as Nintendo 64 pops up in Japan. The success of both Nintendo and Sony in the console front starts weakening Sega's position in the market. The following year arcade games shift more towards sports instead of violence, to the joy of many a politician and parent. Lara Croft makes her debut as *Tomb Raider* and though the game is loved by many, the portrayal of the main character is seen as sexist. The Tamagochi is released. In 1997, a computer beats world champion Gary Kasparov in a match of chess. The Playstation sells its 20th million unit, becoming the most popular game console to date. *The Legend of Zelda: Ocarina of Time*, argued by many as one of the best games ever made, is released in 1998. Sega's Dreamcast also debuts that same year. Much like *Doom* and *Dune II* a few years before it, *Everquest* popularizes massive multiplayer online role-playing games (MMORPGs) in 1999. Finally, the change of the millennium sees the release of both the PlayStation 2 and *The Sims*, the former driving Sega out of the console business for good and the latter enthralling the masses, especially women, with its simulation of real-life up to the point where it becomes the best-selling game of all time. [19, 20]

The 1990s sees the rise and fall of giants as Nintendo solidifies its position in both the console and game market, Sony rises to the top of the pack with its two PlayStation consoles and former top-contender Sega simply cannot, despite a valiant effort, carry the day on the back of Sonic and has to forfeit the console race. Video games become complex, deep and challenging and many a classic is made during this time period, like *Tomb Raider*, *Mortal Kombat, Warcraft, Mario Kart* (1992)*, Half-Life* (1998) and *Final Fantasy VII* (1997). Due to the ever increasing amount of detail in the graphical presentation of games, politicians and parents start taking note of video games and their effects on children. While the foundations of many a genre had been laid before, even more game genres find their flagship game that lifts them to the spotlight. Much like solitaire brought video game to the notice of millions ten years before, *The Sims*

captivates a largely untapped audience resource in women. Also of note is the rapid rise of Internet as a games like *Everquest* or *Ultima Online* (1997) simply could not exist without it. From a purely GUI perspective, it is fascinating to see how much technological advancement increased during this period. At the beginning of the decade PC games were a minor niche and the best console graphics were 16-bit games, with 8-bit consoles and games still being the majority. By the end of it, graphics had gone all the way to PlayStation 2's very recognizable characters, objects and worlds that actually looked very much like what they were portraying and especially cinematic cut-scenes were of incredibly high quality. No other decade boasts a visual development cycle similar to the 90s.

2000-2010

2001 sees the entrance of Microsoft into the console market as it releases Xbox to compete with Sony's PlayStation that has taken over the industry. The console becomes a hit and no small part of it can be attributed to not only the console's successful design, but to its flagship game and arguably the most innovative console FPS game of all time, *Halo: Combat Evolved*. Nintendo releases the hand-held console GameBoy Advance. The following year, games are starting to be taken seriously as the U.S. Army uses *America's Army* to recruit new soldiers and Woodrow Wilson launches the Serious Games Initiative to address more serious issues through games. Valve, the company responsible for one of the most memorable FPS games ever (*Half-life)*, releases Steam in 2003. The digital distribution platform takes the world by storm and becomes the de facto marketing and gaming platform on PC. 2004 sees the release of Nintendo's new hand-held console, Nintendo DS, further establishing the company's dominance in the portable consoles market. Sony tries to compete a year later with the PSP, but even decent sales and much more realistic graphics can't shake Nintendo's grip on the market. Microsoft brings out the Xbox 360, the successor to Xbox and the first console to be able to display high-resolution graphics. Moreover, it brings seamless online play through Xbox Live, a feature which so far had been hampered or altogether lacking on consoles. 2006 sees Sony match Microsoft with the release of PlayStation 3 and continue the rivalry, with the two juggernauts competing for the same audiences. Nintendo also releases its own console, the Wii, but does not aim to directly compete with the other two. Wii doesn't possess the same kind of computing power as the PlayStation 3 or the Xbox 360, but it boasts a new, innovative controller mechanic which allows the users to get up from the couch and use their movements to play games. The motion-sensitive remotes become a huge hit and tap into yet more unclaimed player resources in people of all ages, most notably the elderly, as the family-friendly games are fun, new and exciting to play with the new motion controls. Though *Guitar Hero* brought music games to the masses a couple of years earlier, it was *Rock Band* that hit

the jackpot in 2007 with its full set of guitar, bass, drums and vocals. By 2008, the fact that video games had already become a mainstream entertainment method was beyond doubt as *Grand Theft Auto 4* breaks all kinds of records by selling more than 6 million copies in a week and *World of Warcraft*, launched in 2004, hits its peak with more than 10 million unique subscribers worldwide. The rise of smartphones, Facebook and the ever increasing access to Internet around the globe also sees the rise of social and casual gaming with titles such as *Farmville* and *Angry Birds*. Nintendo capitalizes on the success of Wii by releasing Wii Fit, further incentivizing people of all ages to get up and move with games. Riding on this success, *Wii Sports* surpasses both *The Sims* and *Super Mario Bros* to become the best-selling video game of all time the following year. In 2010, seeing the success of Wii and Wii Fit, both Sony and Microsoft release their own motion-control systems for their consoles, PlayStation Move and Microsoft Kinect. Out of nowhere, Swedish developer Markus Persson's *Minecraft* sweeps the globe and millions of people spend their days hacking away at bricks. [19, 20]

The previous decade sees the established names of the video games industry vie for dominance, with few to none new, real rivals emerging to compete with Sony, Microsoft and Nintendo. Their consoles have been accepted as the platforms for gaming and developers world-wide focus instead on making games for them. The emergence of Steam starts shifting focus away from physical, on-the-store-shelf copies of games to digital versions of them and as Internet is now more or less a given for most, the masses embrace it with little resistance. Graphically, games transform from very good to amazing with the transition from PlayStation 2 and Xbox to high-definition consoles like the PlayStation 3 and Xbox 360. Nintendo keeps doing Nintendo things and revolutionizes the way we play games yet again with motion controls, a trend Sony and Microsoft are quick to follow. Music games are one of the more notable fads of the decade as they are met with roaring success, yet it is relatively short-lived. Lastly, it's the decade that sees the birth of independent game studios and developers as the tools used for video game development become more and more commonplace and people everywhere try their hand at attempting to make their ideas become reality.

2010-present

The first year of the decade sees the release of *The Elder Scrolls: Skyrim*, the fifth installment in the popular series. The game is a massive hit and with an active modding community, still showcases the utmost of what PC games can offer in terms of playability and graphical marvel to this day. Augmented reality begins to rise its head with *Skylander's: Spyro's Adventure* and the soon-to-follow *Disney Infinity*, but it never makes a decisive breakthrough until *Pokemon Go* is released six years later. It does,

however, pave the way for virtual reality which is becoming ever more popular in present day game industry. Ever more game developers turn to their audiences to fund their games and ideas with the popular crowdfunding service Kickstarter in 2012, where anyone can pitch an idea and ask for funds to make it happen. Some of the most notable crowdfunding success stories include the OUYA console and the virtual reality pioneer, Oculus Rift. The year 2013 sees another chapter in the battle for console market supremacy as both Sony and Microsoft release their new, immensely powerful consoles (PlayStation 4 and XBox One, respectively). These consoles have the computing power to produce almost photorealistic visuals even in generic gameplay and also offer even more advanced online and social play functionalities. Not only does the graphic quality take a leap forward, so do the themes and stories present in games, as hit games like *Gone Home, The Last of Us* and *Bioshock: Infinite* demonstrate. Perhaps most notably, the decade sees the rapid rise of Esports with titles like *Dota 2* (2013)*, Counter-Strike: Global Offensive* (2012) and, most recently, Blizzard Entertainment's *Overwatch* (2016) captivating millions of spectators as professional gamers vie for hundreds of thousands, even millions, of dollars worth of prize pools in tournaments across the globe. A great many trends, controversies and stories continue to emerge as the game industry has become a juggernaut in the field of entertainment, on par with any other industry in it. Women's role in games and in the industry, the rise of streamers and youtubers, the simultaneuous popularization and outrage of the free-to-play business model and microtransactions and the rise and fall of indie game developers are just the tip of the iceberg as the video game industry continues to become a great factor in more and more people's everyday lives. [19, 20]

There really is no end to the evolution of the chosen industry here, is there? Just when the big players in the field started stabilizing, if you could even say so, changes shake the way we play and view games. Some youtubers simply play games for the amusement of others and have millions of followers, making a living out of it with ease. Professional gaming, once scorned as a career path with no future, is now making its way into schools. Ever more people are finding work within the game industry as media personalities, hosts, casters and analysts and the biggest video game tournament of the year, *Dota 2's* The International 2016, garnered a prize pool of more than 20 million dollars. In retrospective, it is amazing to see how far the game industry has come in just some 40 short years and, at least in this writer's opinion, it is just as amazing to see what will happen in the future.

3.2 Immersion & suspension of disbelief

The goals that games and game designers want to achieve vary from education to pure fun, but there is a common denominator for every game out there: user engagement and enjoyment. At the end of the day, games exist to entertain the player. Regardless of what type of game it is, regardless of what genre it belongs to, what messages it wants to get across, what subject it revolves around, games are made to be enjoyed and had fun with. This idea of fun has, in the past decade or so, been also making its way to other software and even physical activities as well with the concept of gamification (which I will discuss later on in this thesis) but it is inherent and crucial in video games.

The world is full of different people with drastically different ideas of fun, which introduces a problem to game designers worldwide: how do you design and create a game that can be enjoyed globally, regardless of gender, age, ethnicity, religious views or a hundred other distinguishing factors that can be used to define a person? The short answer is, you don't. You will always have people criticizing your creations, but that simply comes with the territory. All you can do is create stories, mechanics, dynamics and worlds, that follow the “common sense”-design, meaning that many things are fair game in your game if it's believable within the context of the game and is not considered blatantly offensive to a certain community in the real world. Other than that, you just have to roll the dice and hope more people like the game than don’t. When a game manages to make you let go of real-world thinking and accept the boundaries of the virtual world, and the fact that it’s a game, the idea of being offended by something related to you as a person diminishes as in the game world all players tend to be equal and only separated by their skill in the game. Games are universal equalizers as you generally have no idea what kind of a person you’re interacting with and you can only form your opinion of them on how they present themselves. While this can occasionally serve as a reminder of it’s only a game, it can also greatly help enjoying the game and making new, similarly-minded friends as many genres have their own player-bases that at least somewhat think alike and have something in common. Enter the terms immersion & suspension of disbelief. [2]

When you're playing a game, you generally need a couple of things to submerge yourself into the flow of it. A fluid, usable user interface is one. Clear logic and understandable mechanics within the game is another. Causality relationships, i.e. if I kick a ball, it will move in the direction I kick it, within the game is a must. Depending on the type of game you're playing, believable characters, environments, world and story can make or break your game. All of the above-mentioned issues might even be perfectly in order in your game and it might still fail miserably. Making games is never a perfectly surefire art method as you're taking a risk every time you design something,

no matter how safe the design solutions seem to be. Making games is a risky business, but it can also be a calling. [2]

Immersion can be found in a variety of details, as Ermi & Mäyrä (2005) suggest. They claim that player involvement, and by extension player immersion, can be distinguished in three ways: story oriented, game system oriented or sensory oriented involvement. Story oriented involvement (or imaginative immersion) means getting absorbed by the game's story and characters and the lore that is present. Game system oriented involvement could also be called challenge-based involvement as it involves using one's own mental and motor skills to overcome challenges and succeed in goals within the game. Lastly, sensory based involvement revolves around the audiovisual experience of the game and is possibly the most universally appreciated of the three as even observers can participate in it. Even though I will not be using this division of player immersion factors all too much in this thesis, it is an excellent reference point and showcases some of the basic ways in which immersion can be achieved. Naturally there are more factors to making a game enjoyable and immersive, but on a high level the types listed above can help readers understand the different aspects of immersion a little better. [21]

The way the GUI is designed and implemented into the game is one of the biggest factors in generating immersion and encouraging the suspension of disbelief (which basically means that the player accepts things impossible in the real world as possible within the context of the game because of a variety of reasons). How information is displayed to the player is crucial both for giving her the necessary information about what's going on and also for building the experience as a whole. Imagine a racing game where you couldn't see your speed, gears, map, position or route. Even simulators tend to have at least some of these features, let alone more arcade car games, but removing everything and simply displaying the road and the car, while it makes for an interesting thought, would probably not be optimal for the player. Or a FPS where there are no displays on the map and you wouldn't get any feedback about being hit other than simply flopping to the ground unable to do anything. Not fun.

Some genres allow for easier UI integration than others by nature. Science-fiction and fantasy are some examples where necessary information to the player can be quite easily included as part of the game world, or as things that the player understands the characters in the game also see. For example, *Metroid Prime* (2002) or more recently *Crysis 2* (2011) have player characters that wear highly advanced combat helmets and the necessary information is shown as being displayed on their visors for both the character and player to see. This helps create the feeling of being actually there and in the boots of the main character. [22]

Other genres abandon the logic of an incorporated interface as altogether unfeasible such as RTS games or sports games. A great example of including yet not

including the interface as part of the game itself is the *Command & Conquer*-series. In them, the player plays the actual game from a top-down view where the controls are clearly on top of the UI and not visible to the units on the field of battle. However, during the cut-scenes real world actors move and talk about events and missions in the game. The player is addressed directly by the actors speaking at the camera and referring to the player as "Commander". This entices the player to think of themselves being a participant in the scene on a computer screen and an actual character in the fictional world. While the controls and the UI itself are clearly for efficiency purposes and for the player only, the games still manage to create a great feeling of immersion. [22]

There are many other examples of great design solutions for increased immersion, such as the player's health status visible on the back of the character's suit in *Dead Space* (2008) or the main character's sword glowing when a finisher is available in *Middle-Earth: Shadow of Mordor* (2014), both from very GUI inclusive to function-based overlays. The variety of design decisions only shows that there are many ways to create a successful GUI without breaking immersion or hampering user engagement, despite it being a daunting task occasionally. Finding the perfect balance for your game between usability and immersion requires work, but is generally worth the effort.

To analyze design solutions and what aspects weigh more in different contexts a bit more in depth, I will be dividing games into four categories: **casual games**, **simulation games**, **action-based games** and **story-driven games**. A single category can include games from a variety of genres as the idea is to focus on what is important for the player when they are playing a certain type of game, for example a game where they need to actively take actions in the game to survive (first-person shooters, role-playing games) or *Candy Crush/Angry Birds*, where the player simply chases points in a casual manner. Even though some things stay the same, for example the game has to be fun to play regardless of the category, there are major differences as well.

3.2.1 Casual games

Casual games are games that do not require much effort, skills, patience or financial contribution in most cases. They are short, repetitive games that can be played during a short bus trip or covertly during university lectures while the lecturer is not watching. Wikipedia definition for casual games:

"*A **casual game** is a video game targeted at or used by a mass audience of casual gamers. Casual games can have any type of gameplay, and fit in any genre. They are typically distinguished by their simple rules and lack of commitment required in contrast to more complex hardcore games. They require no long-term time commitment or special*

skills to play, and there are comparatively low production and distribution costs for the producer.

Casual games are typically played on a personal computer online in web browsers, although they now are starting to become popular on game consoles and mobile phones as well." [23]

Casual games have permeated the everyday life of most everyone who owns any kind of computer device, be it a mobile phone, tablet or even a smartwatch. Casual gamers are highly popular especially among women and they tend to have a high attraction value even for people who don't normally play video games. Examples of casual games are Facebook games like *Candy Crush* or *Farmville*, mobile games like *Angry Birds* or *Clash of Clans* or computer games like *AdVenture Capitalist* or *Cookie Clicker.*

Figure 1: *In Angry Birds (2009)*, you collect points by sling-shotting your birds into the structures, trying to pop all of the pigs

Casual games get their allure from the fact that you can just pick it up and instantly play it, with no or very little need for tutorials, guidance or setup. Casual games are meant to be enjoyed on the fly and in short bursts. Many of these games are free-to-play games that monetize by asking the player to make micro-transactions in the

game's in-built shop where you can for example buy accessories for your avatar in the game or one-use tools to help you get more points in the next round of the game.

In casual games, story or the narrative environment is very rarely of importance and is generally there just to add some flavor, if they exist at all. When was the last time that Farmville provided you with in-depth lore, that you absolutely wanted to learn more about and went on Google to find additional information about it? Indeed, casual games are the snacks of the video game industry. They focus on easy game mechanics and the speed of results. In most casual games the player's actions have instant results. For example, in Farmville you can gather your products in a matter of minutes, even if in real life crops take months to grow and harvest. To counter-balance the speed in which results are achieved, most casual games implement some sort of stamina or energy bar. Most actions the player takes depletes this bar and once it's empty, you either have to wait a certain time for it to replenish or buy some recharge packages from the in-game store.

Immersion and the suspension of disbelief do not play big roles in the enjoyment of casual games. They are meant for quick and easy fun and are not meant to be immersed in for hours. They are very lightweight in their depth (even though their mechanics can be very complex in some cases) and exist to provide a distraction or some light entertainment for even people who don't otherwise play video games.

3.2.2 Simulation games

Simulation games consist of a meticulously designed game environment, where every button, object and lever in the UI behaves as they would in a real-world environment. Some simulation games take this approach to the extreme, as can be seen from the *Flight Simulator*-series, where the player takes the role of an airplane pilot, sits in the cockpit of various planes and tries to fly them as realistically as possible. Wikipedia definition of simulation games:

*"A **simulation video game** describes a diverse super-category of video games, generally designed to closely simulate aspects of a real or fictional reality.*

A simulation game attempts to copy various activities from real life in the form of a game for various purposes such as training, analysis, or prediction. Usually there are no strictly defined goals in the game, with players instead allowed to freely control a character. Well-known examples are war games, business games, and role play simulation." [24]

Figure 2: In *Farming Simulator 2015 (2015)*, the player is thrust into the life of a farmer trying to plow his trade in a simulated world

Other simulators focus on the mechanics and dynamics present in the activities they attempt to portray. In *Farming Simulator 2015*, the player is thrust into the role of a farmer trying to manage his/her fields. This is done by operating a variety of farm equipment, but rather than giving the player a first-person perspective of the inside of every machine, the player's avatar simply enters the machine in question and the player is then given keyboard shortcuts that activate the different functions of the equipment in question.

Another type of simulation game is the game *The Sims*. In it, players create and take care of virtual avatars in a digital world where everything functions similarly to their real-life counterparts, for example fridges store food and televisions provide entertainment to the Sims. The player is responsible for the well-being of all of the characters in their household and if they are neglectful in their duties the characters become unhappy and unproductive. They can even perish if not given the possibilities to fulfill their basic needs like eating or sleeping. This type of simulation differs quite much from the more realistic simulations mentioned earlier, but nonetheless belongs into the same category as it simulates the causalities and actions of everyday life.

In addition to the above-mentioned simulation types there are a number of others as well, like war and sports simulators, but going into detail about every one of them is out of the scope of this thesis. Regardless of the type of simulation, all simulation games place extreme emphasis on causality relationships and portraying their real-life counterparts as accurately as possible. Simulation games are all about the mechanics and dynamics present in the chosen environment. Every button needs to work as they do in reality, every gauge has to show what it would if the player was actually looking at it

in the helm of an excavator, every action taken has to lead to the appropriate reaction from the environment (I.e. pulling a trigger makes the gun fire). Simulation games place very little emphasis on story-telling or lore and most text that exist within these games is descriptive and factual, not there to spin tales of imagination for the player. Simulations live and die by their mechanics. Even graphical presentation, which nowadays is on a more than adequate level anyway, is a secondary concern.

Figure 3: If you're too lazy to throw parties in real life, you can simulate it with *The Sims*

3.2.3 Action-based games

The first thing to note here is that by action-based games I do NOT mean action games. These things are two separate terms. Action games are defined by constant movement, things happening on the screen and the player constantly dodging bullets and shooting back at his enemies (or something similar). Action-based games, as is the title of this category, should be understood as games where the player has to take actions constantly to influence the flow of the game, be it a FPS where the player is trying to survive a zombie outbreak, a racing game where they are required to maneuver their vehicle in order to stay on the winding race course or a strategy game where the player constantly adjusts tax rates and troop movements in feudal Japan. The category could also be thought as "games that require constant player input" and includes more than just action games.

Action-based games place value on causalities and mechanics, much like simulation games, but they do not emphasize them to a similar degree. Naturally, if you slash at an enemy with your sword, you expect it to respond to the slash with some appropriate action, be it staggering from the blow or raising a weapon in an effort to parry it, but the game is not ruined if the enemy doesn't get a cut in the exact depth and angle of the blow. An approximation is usually enough. Some action-based games take this more seriously than others, but the general idea remains the same. Technically, almost all of the games in the other categories could also be included in this category, but the division made in this thesis serves to focus attention on what things are important in different games for player immersion and enjoyment.

Figure 4: *Dota 2* is a five-on-five MOBA (multiplayer online battle arena), where the objective is to destroy your opponents' main building called the Ancient

In addition to mechanics, action-based games can place varying value on story-telling and believable environments and characters. Some games use it as a kind of a sugar-coating to make players go "Ah, I know that thing/character/object". An example of this is *Mario Kart*, where a number of Nintendo's most famous characters are put behind the steering wheels of various racing vehicles and made to race each other in tracks made in the familiar environments of their previous games. The game at its core is a racing game like many others, but it gets an aspect of familiarity and story by including known characters and environments and, by extension, their existing friendships, rivalries, history and lore.

On the other end of the scale, and very much balancing on the edge of being moved to the next category, are games that use story and plot to give the player directions and motivation to wade through the endless ranks of enemies the game throws at them. These games tend to have cut-scenes, a varyingly well-written storyline

with more in depth characters and character advancement and overall more emphasis on why the player should be doing what they're doing, be it saving the world from hordes of demons or discovering ancient relics from long-lost tombs. Games that use story as an incentive also tend to have voiceovers and dialogue even during player-controlled sections to spice things up and, for example, let the main character crack jokes at incredulous situations or at their enemies. Despite giving more weight to narration and character interaction, these games still put the biggest focus on fluid gameplay and efficient mechanics, as enabling the player to do things is the beating heart of action-based games.

In terms of user interfaces, action-based games value efficiency. Nowhere is this more prominent than in strategy games, where the player needs a lot of tools and information to make the best decisions and act fast, usually with great accuracy, based on those decisions. Even in other games in this category, like most first/third-person shooters or sports games, user interfaces are required to display information, generally in a way that pops out somewhat from the rest of the game so the player doesn't miss anything valuable, like his character's health bar, skill cooldowns or ammunition counter.

Picture 5: Many a friendship has had its strength tested on the colorful tracks of *Mario Kart (Mario Kart Wii, 2008)*

3.2.4 Story-driven games

Despite being closely related to the previous category, story-driven games are separated as their own group as their main driving factors differ somewhat from action-based games. Where action-based games live and die with the fun-factor and addictiveness of their gameplay, story-driven games are all about immersion, suspension of disbelief and eliciting emotional responses from the player. Of course, as in most any

game, the gameplay has to be at least decent for anyone to play the game, but people who enjoy the plot and the journey the game offers are usually willing to forgive some small shortcomings in the mechanics department.

Great examples of story-driven games are many RPGs, action-adventure games and horror games. Naturally there are narratively driven games in many genres, but in the three mentioned genres story can make or break your game and is at the very core of the game experience. Could you imagine a horror game where you're simply getting spooked around every corner with no reason as to who the player character is, where you are or why you are there? Sometimes even the thinnest of storylines can make a difference and become the driving force in your game, like in the popular *Slender: The Eight Pages*-game where you simply have to collect eight pages of scribblings while avoiding the Slenderman who is chasing you. Narratively this is not a very complex setting, but the hallmark of great stories is that where words and descriptions fail, imagination takes over. Simple, yet incredible effective, as countless Youtube-videos can attest to.

Figure 6: In *Slender: The Eight Pages (2008)* you need to collect the eight scattered pages and avoid this very situation.

Story-driven games can occasionally be seen almost as interactive movies. Games like *Beyond: Two souls* or *Heavy Rain* spend a lot of their time showing the player cinematics and conversing characters, much like a tv-show or a full-length cinematic film would. In the player-controlled sections the character usually eventually ends up in difficult situations, be it rather quickly or with a longer build-up through an assortment of ordinary activities that somehow snowball into something dramatic,

where the player has to make huge moral and ethical decisions that can have great effects on the rest of the game, the game's world and the characters.

For example, in the scene shown in picture 7, the player is looking for clues in a murder case. If the player does not find the right clues and enough of them, the case might not get solved which then prevents the character from influencing something further on in the game. In *Heavy Rain*, some decisions or failures can be terminal and one (or more) of the characters the player controls during the game can die permanently, which ripples through to the later parts of the story. The player becomes emotionally invested in the well-being of the characters in the game and cares about what happens to them, which influences their decisions in the game. This is exactly what makes story-driven games a category of their own and give a different kind of experience in comparison to other types of games.

Figure 7: Scene from *Heavy Rain* where the player character needs to look for clues in a murder case.

3.3 Persuasive technology

Video games are not only a great form of entertainment, but also a means to affect players' views and behavior when used correctly (or in some cases, even incorrectly). This type of digital product is called persuasive technology, which can be understood as a piece of software or technology that can change the way the user thinks or acts in the

real world based on their experiences with the technology. Important to note is that the persuasion is done through social interaction and enjoyment of the game, not coercion or deception. [25]

Games are an excellent example of persuasive technology. Video games, especially in the more story-driven genres like RPGs or action-adventure games, tend to offer the players a variety of possibilities, alternative decisions and courses of actions to choose from that have some sort of effect to the characters and the world inside the game. Video game worlds are designed so that player actions have fast-showing consequences and, more often than not, the player can influence everything around him to a great degree. Deciding whether a character lives or dies or whether you want to violently intimidate or gently smooth-talk to someone to achieve a certain goal makes the player decide which sort of person she wants to portray in the game and what kind of moral values they wish to proliferate. [25]

However, seeing the effects of your actions often immediately and in the long term can make a player, either consciously or unconsciously, start wondering about the validity of different kinds of actions as well. An excellent example of far-reaching consequences is the *Mass Effect*-series. The player is thrust into the shoes of commander Shepard, basically the galaxy's only hope of avoiding universal annihilation. This is not a given at the start however, as the player first needs to find out the threat exists and then persuade others to believe it as well so that they might stand together against the enemy. The player has to make a ton of choices along the way, which affect everything from the attitudes of various factions around the galaxy to the player to the very lives of the player's companions. An untoward decision concerning a non-player character made in the first game can be referenced by that same character in the third game as a reason for them not helping the player. Different courses of action produce different results and change how the player's character is viewed. Seeing the effects of your actions in the game often carries over to real world experiences as the player learns about different social interactions and ways of thinking possibly even without realizing it. [25]

Through persuasive technology, designers can have a huge impact to the way others see and think about things. Video games have become so infused into most people's everyday lives that game designers have great opportunities to get their messages across through their games. Granted, many games simply focus on fun and user enjoyment (one can't really argue that for example *Angry Birds* has much of a moral message in it), but it's always an option should the designers choose to take it. It also brings a lot of responsibility not to abuse it. One example of a controversial design solution was the much publicized scene in *Call of Duty: Modern Warfare 2* (2009) where the player briefly takes the role of a terrorist in an airport and is tasked with opening fire at a mass of unarmed civilians. The player is not forced to do anything, but

most of them no doubt went with the approach “it’s a game, I have a gun and the possibility to use it”. However, firing on innocent civilians in a virtual environment is something that causes most people to pause and consider the act, even for a moment. No matter if it’s “just a game”, the brutality of the act effectively breaks immersion and reminds the player that their decisions matter. This is the power of persuasive technology. [25]

Immersion is a key factor in video games as a tool of persuasive technology. When the player immerses herself in the game’s world and becomes emotionally attached to the characters and places around her, decisions that benefit or cause harm to them become personal. For example, if you’ve fought alongside a companion for most of the game and you’ve come to like their witty comments and self-assured manner, revealing them to be a double agent for the enemy at the final scene can be an incredible emotional experience, considering nothing of it was real and it was “just a game”. Regardless, the player experience very much compares to real world experiences and is very important to the player because they have allowed themselves to become immersed in the game world, to basically become the fictional character they were playing. This is exactly what makes videos games amazing in my own experience, as you can be anyone you want and do unimaginable things that simply aren’t possible in the real world, like fly or use magic. Not all games and game genres make use of immersion to the same degree, but those that do (and even those that don’t to some degree) have great potential to form, change or reinforce behavioral patterns in players, either knowingly or unknowingly. [25]

3.4 UI elements in games

There's been a lot of talk so far on games, game design and usability in general, but what about the core subject of the thesis, graphical user interfaces? A bit later on I will dissect the GUI of a few games and go over many of the embedded elements and their function in the game, but for now, let's look at the UI elements on a bit more higher level.

Fagerholt and Lorentzon (2009) proposed an excellent categorization for user interface elements in video games based on how deeply linked they are in the game's narrative and geometry:

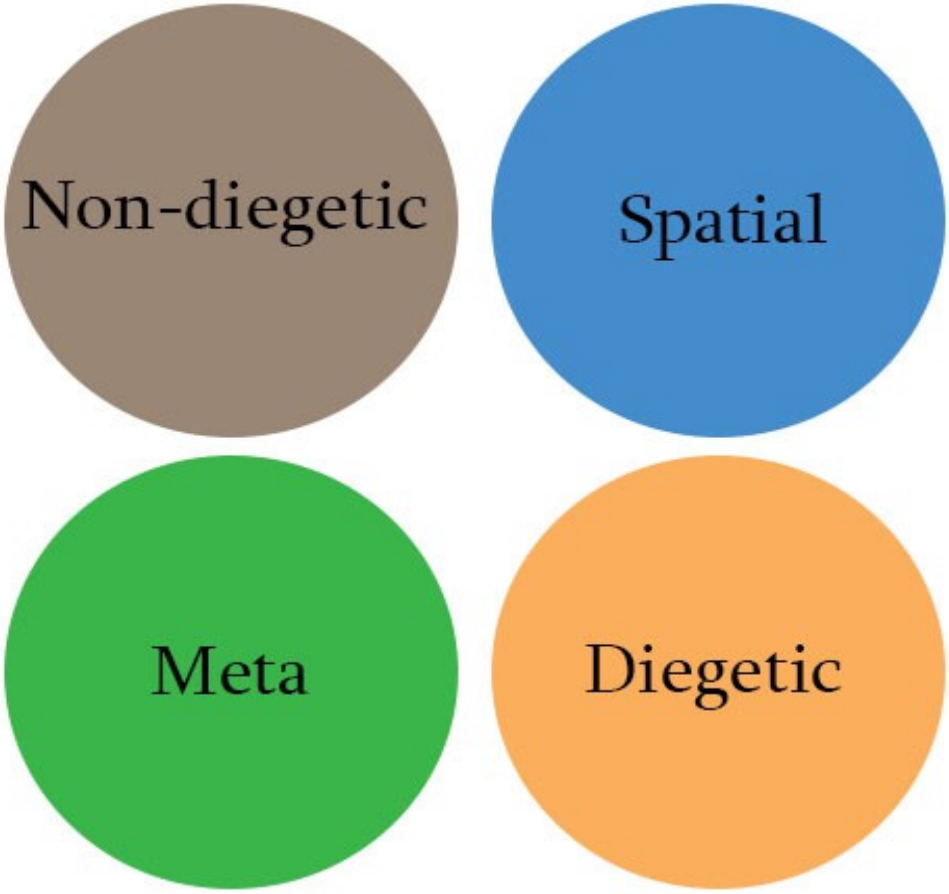

Figure 8: Adaptation of Fagerholt and Lorentzon's categorization of UI elements [27]

In layman's terms, the variety of UI elements in any given game can either be seen by both the player and the player's avatar in the game (diegetic) or understood as invisible to the avatar and only there to provide information to the player (meta, non-diegetic, spatial). Furthermore, elements that exist only to provide information to the player, and which the avatar isn't assumed to be able to see, can be further divided. [26, 27]

Of the four categories, the only one where information is provided in a way that tries to make the player think that the avatar himself is viewing that very same information, is diegetic. More often than not diegetic elements can be found in futuristic games where the avatar may have a helmet or some sort of holograph device, where for example the map or the ammo counter could easily be projected on. Some developers have however implemented diegetic user interfaces in games that portray modern or even historic times, with varied success. [26]

Spatial UI elements try to keep the player immersed in the game as much as possible, even though the information they provide is clearly for the player's eyes only. Tutorials are a common place to find spatial elements as instructions are given in a way

that try to fool the player to think that the avatar is receiving the training, even though the texts and directions are not visible to the avatar. [26]

Meta elements are closely related to spatial ones, but with the distinct difference of sitting on a clearly different plane from the game world. Generally meta elements sit on a 2D plane above the game screen and provide player information on what is going on, for example blood spatters showing up on the screen when the player is hit. Meta elements still try to keep in line with the visuals and aesthetics of the game and don't differ too wildly from the game's content so as to not too noticeably break player immersion. [26]

Lastly, non-diegetic UI elements are there to purely provide usable information visualization and clear controls for the player. They sit on a 2D plane above everything else like meta elements, but they also usually clearly stand out from the main game view. Good examples of these elements can be found in MMORPGs like *World of Warcraft*. [26]

Understanding this division of the elements can greatly help the analysis of UIs in games, as decisions on what type of elements are used reflects on the design decisions as a whole. However, most gamers don't make the distinction and simply take the game as it is, accepting the UI as well regardless of how long it took the designers to end up with the elements they implemented. While categorization is useful for scientific analysis, user interfaces in games, at the end of the day, exist to provide the player as seamless and usable an experience as possible. A good user interface helps the player affect their will on the game world while easing the cognitive loads placed on the player. Imagine a game of any genre which didn't provide you with any information at all. No instructions, no tutorials, no HUD elements. You would have to figure out and remember everything by yourself. While in some cases this can be a game idea in itself, most times such hardcore experience is not desirable and a user interface is implemented to assist the player. The assistance can include controls, hotkeys, status information or a minimap or anything else the designers feel can help the player make the best decisions they can. Allowing players to customize the UI interface and the controls of the game goes a long way to ensure a smooth playing experience for most people, especially in games where accuracy and response times are crucial (like FPS games). [28]

3.5 Gamification

Before there was virtual reality devices, augmented reality applications on every phone like *Pokemon Go*, casual games, and the issue that is the free-to-play-model, there was a concept called gamification. During the 2010s video game industry has seen the rise of even more innovative concepts and terms than perhaps ever before (but considering how

amazingly full of innovations the industry is and has been, that might be a tall claim). Gamification was one such idea and it was taking the world by storm until eventually calming down and dropping from the peak of Mt. Hype to the casual life of being a useful tool. Gamification is to this day, however, one of the better solutions for increasing work satisfaction and overall user enjoyment, not only with digital products but elsewhere as well.

So then, what exactly is gamification? While there are plenty of definitions out there, some including and excluding one another, it eventually comes down to applying designs and lessons gained in game development to other digital or non-digital contexts to make them more playful, game-like and enjoyable. The definition I'm going to use was introduced by Deterding et al. (2011):

"Gamification refers to

• the use (rather than the extension) of
• design (rather than game-based technology or other game related practices)
• elements (rather than full-fledged games)
• characteristic for games (rather than play or playfulness)
• in non-game contexts (regardless of specific usage intentions, contexts, or media of implementation)." [29]

To supplement this, I'll immediately add that games and gamification are not mutually exclusive. Games themselves can be further gamified, an example of which are achievements you can get while playing a game. No one will dispute that achievements, badges and scores are not game-like elements, yet they are not part of the core mechanic of the game and generally do not provide you a crucial advantage within the actual gameplay, if anything at all besides a badge and a title. Furthermore, gamification aims to influence users' attitudes and encourage desired behavior from the perspective of the provider of the gamified experience, so it's not used to just provide an additional layer of user engagement. There is a considerable amount of psychology involved in designing successful gamification. [29, 30, 31]

Though gamification has only seen widespread usage as a concept and a term in the last decade or so, the idea has been around for quite a while, dating back to the 1980s with the advent of frequent flier miles and cereal toys. These can be seen as the very first types of gamification as they rewarded people for doing something and encouraged a certain type of desired behavior, like flying with the same flight company. The first use of the term "gamification" came in 2002 by Nick Pelling when he used it to describe his work of making hardware "more fun". Over time, it has matured to the concept it is today and much like many other evolutions, it failed more than it succeeded for a long time before designers got the right idea of what it was and how it could be used. In 2012, Gartner predicted that within the next two years, 80% of gamified applications would fail because of poor design. Regardless, the industry is predicted to

be worth more than 2.8 Billion dollars by the end of 2016 and more than 11 billion by 2020, so gamification is definitely something every designer should know about. Gamification can be a useful tool in user interface design as well, which is why it is also relevant to mention in this thesis, no matter how briefly. [31]

The relation between UI design and gamification should slowly be revealing itself by now. Simply slapping game-elements on an existing user interface generally results in appalling user experiences. Not all UIs require gamification to boost their efficiency or user engagement and it is crucial for designers to figure out these situations. There must be clear reasons for and tight interplay between the applied gamification and the goals that the user wants to accomplish. After all, gamification relates more to games than playfulness and this ultimately goes back to the definitions of *ludus* and *paidia* (discussed in section 2.3.1). Gamification, even though a lightweight version of games and game design, still has rules, regulations and restrictions to provide the player clear rewards (and punishments). It's not a “do-what-you-want”-type of tool, it's a precisely controlled and designed method of influencing user behavior and providing enhanced user engagement. [29, 32]

If successful gamification can help a product, digital or not, boost its value to customers, how do you then measure success? Gamification as a concept is not the most concrete one of them all, so how can you really know and measure whether your attempt failed or succeeded? The easy answer is through sales figures and overall retention and acquisition of users. However, this does not validate the actual gamification methods used in the product and the end goal of creating gameful experiences, only the overall end result. While this might be acceptable to many product owners, designers would no doubt wish to know whether a particular mechanism or tool they implemented is actually generating user value instead of being used once or twice and then mostly ignored. Gamifying a product is rarely a one-trick-pony and involves more than one added game design element, so knowing what works and what doesn’t is crucial for future designs. [18]

Sometimes gamification can be mistakenly attributed to phenomena or contexts even though it is actually not present. A good example of this is geocaching, a popular game where players have to physically locate caches in the world and log them into the game application. If the player is playing the game and ends up in a park or a sightseeing monument trail, gamification has not occurred as the player is playing the game. However, if the park or the sightseeing trail offers the geocaching game to their customers as part of the allure to come visit that specific location, they are using gamification. The line between what is and what is not gamification can be blurry sometimes as the very sensation of gamefulness varies from person to person. Some people find stock market trading a gameful experience while others do not. Identifying gamification can be an arduous task, but a must if one wishes to understand how to influence users through it or utilize it in designing user interfaces. [18]

To concretely demonstrate gamification, I'll provide an example of it in use by the online retailer Zappos. They have implemented an application called "The Face Game" in their intranet log-in process. When a worker signs in to the system, they are shown a photo of a randomly chosen co-worker and provided several options as to what that person's name is. If the user chooses the wrong one, they are shown the company directory page of the co-worker and given a possibility to get in touch with that person. This boosts social engagement in the workplace and, optimally, figures to reduce the number of unknown colleagues to zero over time. [30]

Gamification is not the topic of this thesis, so I will conclude this section here. As a trend and a tool gamification is an immensely useful thing to learn as it can help boost user engagement and provide additional value for most systems, regardless of if they are digital or not. Gamification itself could easily be the topic of a whole thesis on its own as I've not even touched on the subjects of extrinsic versus intrinsic rewards, triggers, ability and motivation, or the different motivational behaviors that make people take part in a variety of activities. It's mentioned here to provide an additional tool and point of view for game and UI designers everywhere. Ultimately, at its core, gamification is about motivating people to achieve their goals, develop skills or drive innovation through the use of game-like elements in non-game contexts. [31]

4. GUI DESIGN IN GAMES

As has been said before in this paper, designing user interfaces is not a simple thing. In games, this is even more pronounced as games do not have to adhere to traditional UI design guidelines and can wing it in many aspects. Successful UI design in games can often times be a subjective thing, no matter what methods or proven practices you might use.

To further complicate a game developer's job, most people adapt a certain point of view depending on which type of game they are playing. If the game belongs to the story-driven category established earlier in this paper, some people get immediately ticked off if your design solutions break with the immersion of the world. Then again, other people don't care at all, even if the game is supposed to be played for its story before the actual game mechanics. Naturally, there is a sort of a window that screens players based on their own interests. Someone who is highly competitive and likes to play online team games may completely ignore a story-driven single-player sneaking game, where your only goal is to stay alive and running away is your only solution of doing so.

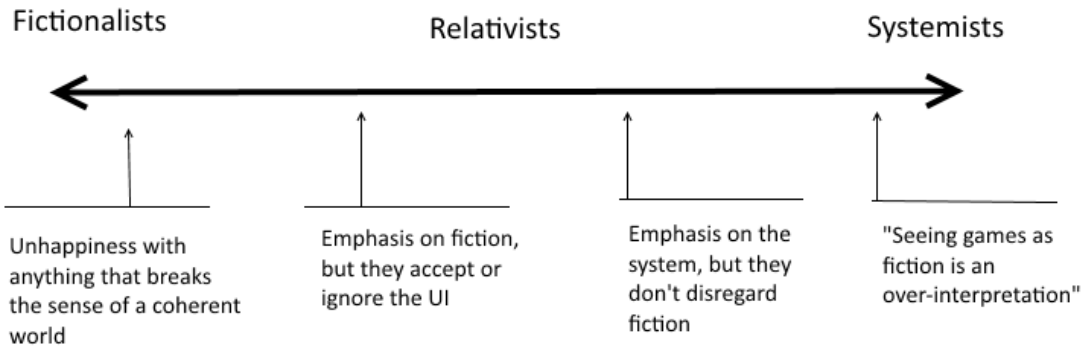

Figure 9: Adapted from Jørgensen [22], demonstrating the division of players

On the other end of the scale are the fictionalists. Anything that breaks their immersion in the game is distasteful to them. They want to completely be immersed in the game and enjoy it to its fullest extent story- and atmospherewise. Opposed to them are the systemists. They see the game world as a fictional overlay for the game rules and mechanics, which are more important to them. Systemists want to master the gameplay itself and simply accept the UI and the game's narrative system as it is without letting it

impact them too much. Between the two extreme factions are the relativists, who portray the compromise between the two. While not squarely placing the weight of the experience to either the immersion or the gameplay, relativists enjoy a mix of the two. Naturally, some people enjoy the fictional part more and some people just want to get good at the game but have an extra lure to play the game by way of an interesting narrative. Moreover, no one is strictly a fictionalist or a systemists but can fluidly move on the scale, depending on the game and even the day in question. [22]

To add even more problems, not all who play your game belong to your game's audience. Any number of people might play your game, even for a while, for a great variety of reasons. Maybe it's on display at a convention or an event somewhere. Maybe someone is showing their friend or relative the game and give them the chance to try it out. Moreover, your game's audience and the people who pay for your game are not synonyms either. Piracy is an ever-present phenomenon in our digitally connected world, but naturally there are legal ways of obtaining a copy of a game as well without actually purchasing it, such as winning a contest or giveaway. Which ones of these user groups should you then focus on, analyze further and involve in game design and feedback? It is entirely up to the developer's own judgement how to handle these questions and there are very few guidelines for these situations. Defending your creative vision can come at the cost of accommodating your audience, and there is very little advice to help you on are you right in any one particular scenario or not. That too makes game development a risky endeavor even at the best times. [33]

4.1 What makes a good video game user interface?

What is it that makes a universally accepted graphical user interface? Isn't good user experience a bit subjective as well? While the former question is the topic of countless debates and the latter's validity is difficult to argue, there are practices and methods which allow you to gain more positive reactions than negative ones most of the time. Game design is never written in stone and all developers need to figure out their own way of solving their specific problems, but you don't have to just wing it every time.

The most simple definition, in my opinion, was perfectly put into words by Desi Quintans in his article about game UI design in 2013:

"Put simply, the role of a good UI is to provide relevant information clearly and quickly, and to get out of the way once it has done its job." [9]

This doesn't mean that the interface simply disappears once the player has got the hang of it. It can if the designers wish it to do so to, for example, increase immersion, but it can just as well remain on the screen to provide further information. The point is, that

once the player has been taught what everything means and what they can do with the UI, it then doesn't draw attention to itself and becomes a tool for the player to interact with the actual game content. Very rarely is the user interface the main point of the game. You don't play a RPG for the cool-looking buttons and character icons, you came to slay dragons and find treasures. [17]

Furthermore, Quintans also provides six excellent questions that all video game UI designers should ask themselves constantly through the development process:

1. Does this interface tell me what I need to know *right now*?
2. Is it easy to find the information I'm looking for, or do I have to look around for it? (Are the menus nested so deep that they hide information from the player?)
3. Can I use this interface without having to read instructions elsewhere?
4. Are the things I can do on this screen obvious?
5. Do I ever need to wait for the interface to load or play an animation?
6. Are there any tedious or repetitive tasks that I can shorten (with a shortcut key, for example) or remove entirely? [9]

While I find these questions to be up to date and crucial for modern designers as well, I would further propose a couple more questions to be added to the list to include other aspects of video game UI design in particular as well:

7. Are the elements of the user interface artistically/graphically in line with the rest of the game? Does anything stand out, and if so, should it?
8. Does the user interface change radically when moving from screen to screen or when opening any of the windows or tabs in it? Are the same elements always in the same place and how do additional information pop-ups fit into the screen?
9. How consistent and efficient is the navigation within the interface? Is there anything unneccessary that's only there for show and could become a liability in the long run when players appreciate efficiency over effects?
10. Does the player absolutely need all of the items on the screen? Is there a better way to do/show/display something? Should anything else be shown/hidden in each of the views and tabs?
11. Have you considered people with some sort of disability, for example color blindness? Is your game up to standards in terms of accessibility?

As Quintans also suggests, ask these questions frequently when you design and play your game. Generally it's not healthy to obsess over many details of your game, but the world of UI design is one of endless nitpicking. When it comes to the user experience of your game, arguably the most important part of it, obsessing about it is fine. [9]

Good user interface design more often than not translates directly into better

user/player experience while playing the game. After all, it is their single-most important tool and the only medium between the player and what's happening on their screen. If the player has to waste time on clicking through unnecessary tabs or dialogues or waiting for an animation each time they open a menu, it can start to take a toll on their enjoyment, especially with longer-lasting games.

Sometimes, however, you can bend the rules and either salvage an ailing design or improve on a good one by allowing players to customize their own UIs, as is the case with *World of Warcraft*. Letting anyone with the skills and tools to mess with your game can be a daunting idea, but it can also pay off big time. Players tend to want to improve the games they like, which benefits the developer as well as long as they serve as an adjudicator, meaning the players cannot create elements that give them unfair advantages over other players. It's a great tool in any developer's arsenal, albeit one that requires careful consideration, a large enough playerbase and constant supervision if utilized. [10]

Another thing to keep in mind is that even if you design the world's best user interface for your game on PC, mobile or console, it doesn't mean it translates well onto other platforms. How the players use different platforms varies and especially the controller scheme is different for each one. Consoles tend to have controllers that also differ from one another a little bit, computers are operated using a mouse and a keyboard and mobile devices have a whole other controller method in touchscreens. All of them bring their own possibilities and challenges which need to be considered individually. Bad ports from platform to platform have ruined reputations and game experiences before, so don't take it as a given. From a graphical UI perspective, the graphical detail also needs to be adjusted, as mobile devices have much less room and computing power for visual glamour than consoles or computers.

Overall, pay especially close attention to parts of the UI that are being constantly viewed or showed to the player. In the case of *World of Warcraft*, most of the fan modifications revolve around the interface bars, the map and inventory management as well as raid tools as these are visible every time you open the game or want to get anything major done. Another example can be found from *Call of Duty 2*, where an indicator lets the player know whether their character is standing, prone or crawling. This is somewhat of an outdated method of displaying information, but some sort of version of this information tends to be a part of all similar games even today, be it a clear indicator or something more subtle. In these cases you need to be wary of what is absolutely necessary and what can be distracting. Something that doesn't bother the player immediately might become annoying over hundreds of times of seeing it. [10, 34]

Even though accessibility is equally a game design issue as well as a user interface issue, it deserves a special mention in this list and this section. Persons with some form of disability or impairment should always be taken into account when

designing a game, as they probably require unique design solutions to be able to enjoy the game. The first example I found was a person who had lost an arm and wanted to play *Diablo*. The game allowed him to do this as the control scheme could be operated entirely with the mouse without having to use the keyboard too much. Naturally, this can't be applied to every game or even every genre, but it is still something that is overlooked alltogether too often. [35]

4.2 Guiding & fallen stars

Games, UI & UX design and good design practices have been talked about in some detail and the next logical step is to delve deeper into existing video game user interfaces to find both success stories and let-downs. This section will provide examples of both good design solutions that won over players and bad decisions that, while perhaps did not alienate a large amount of the game's playerbase outright, definitely weakened the gameplay experience for many. The section mixes both my own experiences and notions from the sources mentioned freely throughout it, both of which are at least a bit subjective, so do not consider everything as absolute truth, just examples.

Figure 10: The inventory screen of *The Elder Scrolls: Oblivion*. Original on the left, modded one on the right.

The Elder Scrolls: Oblivion was released in 2006 and, despite being a commercial success and overall a great game, it had some downright atrocious design decisions in it. Above in figure 10 is displayed the original inventory UI of the game. Considering that the player can have 200 or even more items in the depths of their tunic pockets, using a minimal amount of screen space to display SIX of them at a time is absolutely ridiculous. It took players forever to browse through their items, even if they could use the tabs on the lower part of the inventory to limit their search to just a particular item type like weapons or armor. In the very same picture, on the right side, you can see a modder's version of the inventory screen. It is much bigger and displays a lot more of the inventory in one screen, eliminating the need to constantly click to the next page to see more items. This simply goes to show how even the biggest, most

successful games are not immune to failed decisions when it comes to UI or game design. [9]

Another example is the battle UI of the *Total War*-series. While generally good and informative, the movement controls for your own units can be extremely annoying. As an example, below in picture 11, you can see a screenshot from *Shogun 2: Total War* (2011). While it has no doubt been a monumental task to iterate the battle interface into what it is today, some annoyances can still be found. When you have a large number of individual units to command, it can become arduous to move them in exactly the way you want to. Especially when the fighting starts, holding down your mouse's right click and then dragging away from the click-point to align your troops can not only be a feat in itself, but the icons can get lost in the terrain or underneath the troops, forcing you to cancel what you were doing and start again. This would be an easy fix if the selection color for the units and the location icons when trying to order a movement weren't the exact same color, for example. It can be frustrating having to either pause the game constantly or have to repeat the same action multiple times before getting the result you wanted.

Figure 11: Overview of *Shogun 2: Total War's* battle UI

Sometimes less is more when it comes to providing visual information to the player regarding the character's status or the tools at their disposal. *Call of Duty: Modern Warfare* (2007) had a traditional user interface when it comes to FPS-games: a minimap, scoreboard in multiplayer, hotkeys for additional items the character has and counters for ammo and grenades. What is noticeably missing is a health bar. Whenever the character gets hit, a red haze starts forming on the player's screen to indicate damage taken. The greater the damage, the greater the colorization. This is nowadays a very commonly used method of displaying the character's health situation as it removes a

static UI element from view and only displays the information when necessary and does so in an informative way. The game also features a hardcore-more, where all the hotkeys and counters are also removed from the UI. This feature showcases that even information that you thought was absolutely crucial is sometimes falsely thought of as such. It brings a whole new level of immersion and reality when you actually have to be mentally aware of your clip situation instead of just casually checking a number from the bottom corner of the screen. [26]

Figure 12: The UI from *Call of Duty: Modern Warfare*. Normal view on the left, damaged view on the right.

Another great innovation within the FPS-genre comes in the form of visually integrated counters that I've mentioned earlier in this thesis as well. A great example of this can be found on some weapons in *Doom 3* (2004). Instead of displaying a separate ammunition counter on a non-diegetic plane above the rest of the game, the counters are included in the weapons themselves in a spatial format so both the player and, immersionwise, the character can be understood as being able to see the counters. This provides great immersion for the player and does not break the unity of the UI design if implemented properly.

Figure 13: Screenshot from *Doom 3* with an integrated ammo counter

4.3 Heuristic analysis of video game UIs

Finally taking this thesis into its practical part, I will now demonstrate the analysis of a few different types of games through the questions listed in section 4.1. I will use them to analyze three games that I've recently played and which are somewhat known to the wider gamer audience as well. Bear in mind that the following is my own analysis based both on the aforementioned guidelines and questions and my own experience in the field. Some may view some aspects of what follows differently.

As a sidenote, the list of questions for GUI analysis focuses on exactly that, disregarding most other aspects of usability and game design in favor of a focused point of view. There are other guidelines that can be used to evaluate usability (for example, the legendary Nielsen's usability heuristics) or even similar points for games specifically, but the focus here is the graphical user interface so I won't be going through them. This is not to say that for example Nielsen's heuristics couldn't be used to analyze user interfaces successfully, quite the contrary, but the point here is to use a method specifically designed for this purpose earlier in this thesis. [36, 37]

These three games were selected for analysis based on the author's recent, considerable experience with them. This was a necessary condition for in-depth analysis as otherwise it would be based on vague memories and assumptions which could be very wrong. They all represent a different genre, and do so pretty well, and especially the first two, *Skyrim* and *Overwatch*, are very well known and extremely popular. All

games were played and checked during their analysis to check the insights gained with the design questions. The screenshots shown are all, with the sole exception of the first half of the very first figure in section 4.3.1, taken by myself while playing the games. In most cases I also browsed through Steam reviews and internet forums to see if my views were supported even a little so as not to focus on irrelevant issues.

4.3.1 Case 1: The Elder Scrolls: Skyrim

The successor to *The Elder Scrolls: Oblivion*, which was mentioned as a warning example of UI design earlier in this thesis, *The Elder Scrolls: Skyrim* (2011) improved on nearly all aspects in comparison to its predecessor. Both the players and the critics welcomed it with open arms and it is still one of the most played and talked about games of the past decade. Prolonging its life is the fact that fans have been allowed to customize and create modifications for it (on PC) pretty much as they wished. So if you absolutely want to ride a My Little Pony-horse into battle against an evil, rainbow-spitting Thomas the Train Engine while brandishing Frodo's glowing sword from The Lord of the Rings, you can certainly do that. The game can be played from a third-person or first-person perspective and you can switch between them as you wish. As an open-world game with immense player freedom, *Skyrim* is very much a paidia-type of game with only nominal quest objectives that the player can choose to complete whenever they wish or not at all.

Figure 14: The general UI of *Oblivion* on the left, *Skyrim* on the right

Starting off from the general user interface, elements have mainly been shifted around but as a whole not much has changed. The main difference is that *Skyrim* only constantly displays the navigation bar on the top of the screen and hides everything else while *Oblivion* also had information about the character, like health and the equipped weapon, always visible. *Skyrim* displays the three important bars (health, stamina and magicka) whenever they are lost or gained. The decision removes suprisingly unnecessary elements from the view and the standard view of the UI is actually really

immersive with only the navigation bar being a meta/non-diegetic element on the screen.

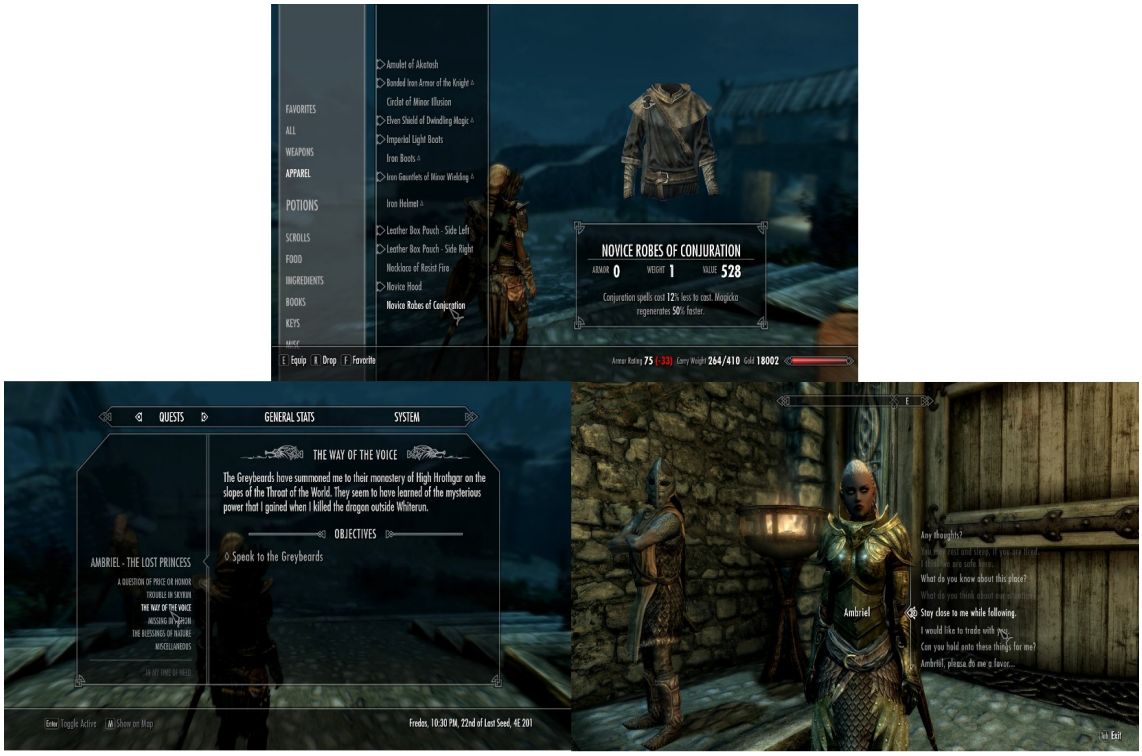

Figure 15: Other views from *The Elder Scrolls: Skyrim*. Inventory on top, quest info bottom left, talking to someone bottom right.

Learning from their past fumbles, the developers fixed the awful inventory view of *Oblivion* and *Skyrim* displays a lot more than just six items at a time. The same goes for quests. Socializing options are also easy to scroll through, though all of these views still do have some issues. Considering most everything can be eased/fixed with mods, it's a wonder that Bethesda, the game's developer company, hasn't fixed them in the base game.

That's about enough warming up, time to work through the list.

1. Does this interface tell me what I need to know right now?

In this regard, *Skyrim* succeeds. Even though most of the information is hidden away and either needs to be opened via a hotkey or shown when it's being influenced, the important information is there when you need it and the exclusion of information from the regular view goes a long way in creating immersion and flow for the player. This is the case in all views, the necessary information is usually available when the player needs it and the minimalistic UI design appeals to fictionalists, relativists and systemists alike.

2. Is it easy to find the information I'm looking for, or do I have to look around for it? (Are the menus nested so deep that they hide information from the player?)

Every menu or screen has a hotkey, although finding specific information, like the stats of your currently equipped weapon, requires the player to click through a couple steps in the inventory screen. Some things, like what is the maximum stamina or magicka value of the character, are in a bit of a weird place (skills overview) as the game does not have a "traditional" character sheet for listing the avatar's attributes. No menu is deeper than a few steps however, so getting lost should not be a problem.

3. Can I use this interface without having to read instructions elsewhere?

If you drop a random player into *Skyrim* without going through the tutorial or previous experience of similar games, the UI could be quite challenging. In this case, the minimalism could work against the user experience as the player might not want to just hit buttons to figure out what they do in fear of doing something bad or something they don't know how to stop. Very few would probably manage to actually progress with the game's storyline as they wouldn't be able to understand what they were supposed to be doing, why they should be doing it or how can they get to the quest location. The menu does have a section for controls, but then you are reading instructions from somewhere. Overall, though the basic interaction and controls could be figured out with some trial and error, much of the game could be missed as the UI is not the most intuitive for a completely new player.

4. Are the things I can do on this screen obvious?

Looking at the general interface of the game, the short answer to the question is a definitive no. There are no indications as to what the avatar is capable of nor what they could do in the game's context. However, the longer answer to the question is actually yes. The only clues are the things on the avatar (if playing from the third-person view), so you might figure out that you can hit, block and shoot things with arrows, or even cast spells. While that's not much, it is absolutely everything you can do in the immediate situation shown in figure 14. When the avatar moves, other actions become available and a pop-up will tell them that they can talk to this person by pressing a specific button or loot the chest in a similar way. The same goes for most other views shown in figure 15. Leveling up is one of the exceptions as the game only tells you that you have enough experience to level up but doesn't guide you in any other way with it. While the UI is distinctly non-informative, it is still usually informative enough.

5. Do I ever need to wait for the interface to load or play an animation?

Skyrim is all about animations when you break it all down. You're watching an animation whether you attack, cast a spell, run, jump, talk or even browse your inventory or skill trees. Most of them are brief and especially the ones relating to the avatar's physical actions are generally understandable, but for example the animation in

moving around the skill tree section can get really annoying really quickly. You're always wasting just a tiny bit of time watching your screen move from one place to another and when you just want to get something done, it can be irritating.

Figure 16: The skill trees in *Skyrim*. There are many of them and moving between and especially in them is quite cumbersome

6. Are there any tedious or repetitive tasks that I can shorten (with a shortcut key, for example) or remove entirely?

Very few things are incredibly repetitive in *Skyrim*, but using items or crafting things can be such from time to time. Though you can tell how many of a certain item you want to sell at a vendor for example, you need to click multiple times on potions to drink enough of them to heal or on crafting targets to make more of them, even if you have ingredients for more than one. This is repetitive and could be easily fixed with a simple script, but they still exist in the game. Especially during crafting creating the materials can be tedious if you need 100 iron bars and have to click to create every single one of them manually.

7. Are the elements of the user interface artistically/graphically in line with the rest of the game? Does anything stand out, and if so, should it?

Visually, no single piece of the interface stands out more than it should for deviant appearance or functionality. The navigation bar is clearly only there for the player to see and not for the game's characters, but it does not disturb the player's immersion as navigation tools are always required and *Skyrim* offers one that fits into the overall style well. Even the menus, being very basic with a darker shade background and simply

lighter colored text, serve their purpose well enough as they are even more obviously non-diegetic elements.

8. Does the user interface change radically when moving from screen to screen or when opening any of the windows or tabs in it? Are the same elements always in the same place and how do additional information pop-ups fit into the screen?

Most of the pop-up windows and information screens are opened on top of the normal game interface and include some transparency, meaning that the player is almost never taken away from the one basic user interface. Pieces are only added to and removed from it. The one exception to this rule is the skills tree, again, which opens up into a whole new screen. Aesthetically it is still in line with the rest of the game, but it also differs a lot from every other view in the game. It is not too confusing for most players and you can always back out of it using the game's regular hotkeys.

9. How consistent and efficient is the navigation within the interface? Is there anything unneccessary that's only there for show and could become a liability in the long run when players appreciate efficiency over effects?

As mentioned, the player is staring at the same interface view for most of the time. When you open up your inventory or list of favourited weapons and spells, the menus open up on top of the normal view and the game is paused in the background. Navigation in these menus is quite clear, with the player moving from category to category or activating one of them, moving to whatever that particular menu option includes. Even the skill tree view follows the same logic, so it is very consistent through the game. However, nearly everything includes some sort of animation, be it however small, which can become cumbersome in the long term. When the player has become an expert in the game and navigating within its menus, she values efficiency over effects that have long since lost their glamour. Forcing the player to wait for the animations to finish every time is an incredibly outdated design solution and should be put to rest.

10. Does the player absolutely need all of the items on the screen? Is there a better way to do/show/display something? Should anything else be shown/hidden in each of the views and tabs?

This is clearly a question that the developers asked themselves between the release of *Oblivion* and starting the work for *Skyrim* as the general user interface can attest to. Even elsewhere, the game is the very champion of minimalistic UI design and it works to create an atmosphere of hardship, struggle and being there for the player. The one thing I still miss is a traditional character sheet with all the basic information about the avatar in it even though the game doesn't have anything like ability characteristics or skills that could be activated after learning them (shouts, or thu-ums, and spells aside of course). There is still plenty of information to display on such a sheet and it would wrap up information about the avatar nicely in one place. Now it tends to be quite scattered.

Also, managing items and equipment can still be a pain, there are definitely better ways to go about both.

11. Have you considered people with some sort of disability, for example color blindness? Is your game up to standards in terms of accessibility?

Skyrim unfortunately is one of the games where accessibility options are very limited. You require both hands to play because of the control scheme both on PC and consoles and the game does not provide too in-depth customization options for other kind of disabilities either. Fortunately a lot of the features in the game can be enjoyed universally, as for example you can turn on subtitles for everything in the menu and though some items and abilities have their own unique appearance and color, there is also a text-based explanation of what it is and what it does. Not everything can be explained like this, for example if you happen to trigger a trap in a dungeon the first (and usually only) warning before impact is a sound. Nonetheless, there are many games out there that fail much worse in terms of accessibility than *Skyrim*.

Overall, *The Elder Scrolls: Skyrim* survives the inspection with good grades. Despite being an incredibly massive game, the basic user interface elements are equally incredibly simple and minimalistic without sacrificing their information value. There are still things that could be fixed, especially the animations in every single menu and screen, but the game can be enjoyed as it is for hours and hours and the player can easily become immersed in the beautiful world that is Skyrim.

4.3.2 Case 2: Overwatch

The insanely successful game company Blizzard, who are responsible for such classic game franchises like *Warcraft*, *Starcraft* and *Diablo*, have also recently ventured into the realm of multiplayer FPS-games with their newest game title, *Overwatch* (2016). The game has quickly risen to become one of the most played games in the world and since its release it has been a popular competetive game in the professional scene as well. It pits 2 teams of 6 players against each other in a range of maps with focused objectives with a variety of heroes to choose from and game mechanics to learn if you want to be great at it.

Much like *Skyrim*, *Overwatch* has one main user interface that the players look at during gameplay. This is the main way the players interact with the game and each other, with the other views mainly relating to menus, navigating them, selecting a hero to play or viewing the skills of your particular chosen hero. Again, despite being a very intricate and finely crafted game with plenty of content, the user interface is very well designed and tested.

Figure 17: The general user interface of *Overwatch* that players look 90% of the time. It can get crowded when everybody presses their buttons simultaneuously.

1. Does this interface tell me what I need to know right now?

Focusing on the gameplay user interface, shown in figure 17 above, the first reaction is that there is a lot of stuff flying around on the screen. When 12 people all activate their heroes' abilities at the same time and move and jump around while doing so, it can get very confusing and fast-paced even for an experienced player, let alone a brand new person looking at the screen. However, when you start drilling into the UI components, you see that it is very well thought out. The constant elements that are always visible to the player are his hero icon (bottom-left corner), their health, ultimate status (bottom middle), the location of the current objective, the skills they have at their disposal, and their cooldown status (bottom-right) and their ammunition status. In addition, on the top, we can see the game's performance scores (top-left, can be turned off), time remaining on the map and the objective (top middle) and the situation with the objectives (below the timer, in this case the attacking team has taken point A and point B is active). When required, a chat window displays information above the player icon on the left edge, a kill feed appears on the top-right corner and information about what's happening is displayed in the middle of the screen (“Objective defense”, in this case). While this may seem like a flood of information, especially when compared to *Skyrim*, for example, we have to remember that this is a whole different kind of game. While a kill-feed or the hero icon are things that are not absolutely necessary for the player, everything displayed on the screen provides information to the player and enables them to make better decisions, sometimes with very little time.

2. Is it easy to find the information I'm looking for, or do I have to look around for it? (Are the menus nested so deep that they hide information from the player?)

Stepping back from the regular gameplay UI, we can take a look at the other views the game provides especially for searching for information and adjusting the settings and hero setups.

Figure 18: Different information screens in *Overwatch*. Hero skill overview on the top, career overview bottom-right and the end screen after a match on bottom-right.

While the normal gameplay UI is quite cramped and condensed to provide incredible amounts of information with the least distractions and unnecessary elements possible, the menu screens don't have similar restrictions. Free to use the entire screen as their playground, the information views contain very much data, sometimes even unnecessarily so. The good thing is that most bits of information are just a click or two away from the main menu, unless you want to find something very specific, like how many hours did the player play on a specific hero during the first competetive season. Then you have to click quite a few times for that information. This is somewhat of a double-edged sword, as the information provided by the game is of good quality overall, but the sheer amount of data displayed can also get in the way.

3. Can I use this interface without having to read instructions elsewhere?

Most views in the game are pretty self-explanatory and give the player advice or labeled buttons to interact with them. The exception to this is the gameplay UI itself. While abilities have their hotkeys displayed, nothing tells the player how to move, jump, crouch or do their basic attack. Granted, most player's would then start dabbling with

the keyboard and mouse and quite quickly find the correct keys as they are very standard to most FPS-games (and can be checked/modified from the menu). Overall, the game's interface views are very intuitive to even an unexperienced player.

4. Are the things I can do on this screen obvious?

Much of what was written under the previous question applies here as well. *Overwatch*'s interface screens are meticulously well thought-out and usually provide instructions for the player as well. Every interactable button and elements is highlighted in some way when you hover the cursor over them, which tells the player that this element can be clicked.

5. Do I ever need to wait for the interface to load or play an animation?

Unlike *Skyrim*, *Overwatch* does not have too many animations in its effiency-based views (like menus). Yes, there are some short transition animations, but they do not bother the player nearly as much as *Skyrim*'s did and don't make interacting with the menus nearly as cumbersome. When the player clicks on something, the corresponding action happens immediately and the player is usually free to do whatever again. The one minor exception to the rule I've found relates to the hero selection screen, where the animation of displaying the currently highlighted hero doesn't always work immediately, but takes a while to load. This does not prevent the player from browsing the heroes however, so it's more of an artificial bug than anything really hampering.

Figure 19: The hero selection screen.

6. Are there any tedious or repetitive tasks that I can shorten (with a shortcut key, for example) or remove entirely?

So far, I haven't found any problems that could be put into the unnecessaringly repetitive-category. Naturally, you repeat a lot of your actions in the game itself, but that doesn't count as you're playing the game. That's what you're supposed to be doing. Here, let's view the main menu:

Figure 20: The main menu

Everything you need is there and you can get anywhere with just a few clicks. At most you need 3-4 clicks to get to the bottom of any single menu option from here. If you are looking to play, you need exactly 2 clicks of the cursor to start searching for a match. If you're playing just some casual games, you don't even need to press the “play”-button more than once, you're automatically put into queue after each match. The only repetition that could be thought of as irritating is opening lootboxes if you happen to have collected many of them, but even that is a minor thing and doesn't really make the game experience worse. The quality of rewards from the boxes does, but that is another thing and doesn't relate to the user interface in any way.

7. Are the elements of the user interface artistically/graphically in line with the rest of the game? Does anything stand out, and if so, should it?

Blizzard did a beautiful job with the appearance of *Overwatch*. The game is full of colors and rich with different characters and their wardrobe options called “skins”. *Overwatch* has a very distinct and unique style and look and every view in it adheres to similar design solutions and colorization patterns. Colors are used to differentiate enemies from allies, the reward qualities from lootboxes and effects in the game. Some

skins have a distinctly different look as they might relate to a specfic yearly event, like Halloween, but they are incorporated seamlessly into the theme of the game. The players can buy golden weapons for their characters with points earned from winning in the competetive mode and those are perhaps the most stark contrast to the regular appearances in the game, but they are meant to be so as they symbolize that player having been successful enough to flaunt it with a shiny new weapon skin.

8. Does the user interface change radically when moving from screen to screen or when opening any of the windows or tabs in it? Are the same elements always in the same place and how do additional information pop-ups fit into the screen?

The contrast between the gameplay UI and everything else is stark indeed, but careful usage of visualization and simply after a few times of getting used to it, the menus and information screens fit in well. It helps that while the main menu is the first thing the players see (after the loading screen) and it clearly adheres to the menu design of the game, it also already includes a random map in the background and a random hero fidgeting about. The view changes pretty much when jumping from screen to screen, but all of the views serve their own purpose and have some consistency between similar screens.

9. How consistent and efficient is the navigation within the interface? Is there anything unneccessary that's only there for show and could become a liability in the long run when players appreciate efficiency over effects?

As mentioned in the earlier analysis points, the navigation within the interface is a few clicks deep at most, usually a click or two less than that. Every menu has guidance of how to get back to the previous screen or what elements you can click to proceed further or alter the data you're shown. While the navigation does have brief animations all over the transitions, the information is displayed very fast and doesn't bother the player nearly as much as the clunky data views in *Skyrim* for example. Naturally, if you absolutely know the exact piece of information that you want and it requires a few clicks, wasting that 0.2 seconds on waiting for the animation can be mildly frustrating, but you're only doing it the one time, maybe two or three at most, so it doesn't really get under your skin.

10. Does the player absolutely need all of the items on the screen? Is there a better way to do/show/display something? Should anything else be shown/hidden in each of the views and tabs?

The gameplay interface is solid, as are most of the menus. The developers clearly iterated through possible designs quite a few times to end up with the almost minimum of distractions on the screen and the most logical presentation of data in the menu screens. However, there are still things that are not 100% necessary to display (like the hero icon in the gameplay UI) or elements that could be presented in a different way

(some of the menu layouts or the information in them), but those are semantics at the end of the day and is more or less subjective to each player. All in all the interfaces serve their purpose admirably without cluttering the screen too badly.

11. Have you considered people with some sort of disability, for example color blindness? Is your game up to standards in terms of accessibility?

Without a doubt, *Overwatch* is a colourful game. If you have some sort of color blindness or eye-sight impairment, you will be at a minor disadvantage. For example, if you have trouble with the color red, you wouldn't be able to see the outlines of the enemy team and would blindly have to guess whether someone is on your team or not (of course their shooting at you might tip you off). This, however, is not the greatest problem here and the game actually offers possibilties to customize the colour patterns for exactly this reason. If you have trouble with your hearing, that's when the major disadvantage kicks in. Footsteps are incredibly important in the game as you can hear whether an enemy is nearby by the sound of their footsteps. Not only that, all of the most powerful abilities in the game, known as ultimates, make a distinct sound which is usually the first thing most players recognize their activation from (for example, the same ultimate ability has a different accompanied voiceline depending on which team the character using it is on). Voice communication is also something that is much more informative than stopping and typing things with your keyboard. There are no options to do anything about this either, for example an option to visualize sound effects would greatly help to counter this problem. The can't also be played with just a mouse or with one hand in any way as the movement and character actions require both hands, good reactions and eye-to-hand coordination to utilize.

4.3.3 Case 3: Darkest Dungeon

Perhaps somewhat the odd man out of the trio analyzed here, *Darkest Dungeon* (2016), despite being a huge success in critic and player reviews alike, is not a game of the same magnitude and expectations as the previous two. Developed by only a handful of people, the game has sold more than a million copies since its release and brings the traditional dungeoncrawler back to modern day players with an excellent showing.

Somewhat unlike the other two games, *Darkest Dungeon* has a variety of different UI views that all serve a particular, important purpose. While the “actual” gameplay happens in a single fixed gameplay UI, the player will find themselves quickly overwhelmed and out of options if she only constantly focuses on getting to the gameplay as quickly as possible.

Figure 21: Hamlet, or ”management”, overview where the player prepares his heroes for the next adventure

Figure 22: Combat user interface, or the gameplay UI. This is the core of the game and what the player usually is staring at.

1. Does this interface tell me what I need to know right now?

The interface views in *Darkest Dungeon* are generally very tailored to suit a particular need and while that makes for quite a bunch of interface views, it works. For example,

the hamlet overview, displayed in figure 21, doesn't exactly tell any specific information about a lot of things, it only shows the heroes on the roster and their overall status in terms of stress, level, class, activity and equipment quality. Then again, the amount of information available could not possibly fit into a single screen, so dividing it up into many smaller pieces is a good call. Even though the UI might look like a village, it is in fact just another tab-based solution for information management.

The combat interface has the ungrateful task of trying to be as informative as possible while still not getting in the way or becoming too cramped. The developers have managed to succeed in this difficult task, however, with a clever mix of spacing, effects and clickable items. The end result displays everything the player needs to navigate a dungeon and survive it, from his inventory to a minimap and heroes' skills.

2. Is it easy to find the information I'm looking for, or do I have to look around for it? (Are the menus nested so deep that they hide information from the player?)

As mentioned, most of the information is hidden away under some sort of tab. The main exception to this is the combat interface, where everything the player needs for combat is displayed, the only tab available being the window that can be switched between the map and the inventory. *Darkest Dungeon* does a remarkable job in reducing the amount of clicks the player has to make to find any information they want to just two or three at maximum. This is no small feat considering the depth and amount of data the game has to offer. Then again, very little information is generally available right at the start for the very same reason, but this is an acceptable trade-off.

Figure 23: A single hero's skill panel.

As can be seen from figure 23, shoving all of this information to the overview screen for all heroes would be absolutely impossible, but as a separate tab it works

perfectly. All the information about any single hero is just a click away and while it does take a bit of getting used to all the icons and texts, the interface does seems like very thought-out solution for this particular information.

3. Can I use this interface without having to read instructions elsewhere?

Some of the views are not very beginner-friendly. Going back to figure 23, the window displays a lot of information and contains what the player might only assume are buttons (hero skills). They cannot be pressed in this particular scenario, however. Each hero can have 4 combat skills active and since no more are unlocked, they can't be changed. This can be misleading to some. Similar situations play out in more windows as the player simply has to learn by trial and error what he can interact with immediately, what he can interact with later and what he can't interact with at all. More complications arise as in some screen you can do something which then enables you to interact with elements that you otherwise can't do so with. The game definitely is not the most beginner-friendly, but once you've learned about what everything does, you begin to appreciate the informativeness and efficiency of it all.

4. Are the things I can do on this screen obvious?

Every window in *Darkest Dungeon* serves a specific purpose. This also enabled the developers to provide instructions for the player in all of them for the specific action in every new view. It is somewhat of a beautiful contradiction that while the UI can become overwhelming and confusing for newer players, it is also simultaneously incredibly efficient and purpose-driven.

Figure 24: Ability trainer where you can improve your heroes' skills

A good example can be seen in figure 24. Despite the screen being quite full of icons and text, there's only a few things to be done here. The window instructs the player to drag a hero from the roster to the spot reserved for him/her in the window which then opens up their skills as seen in the picture. The cost of upgrading a skill is displayed and when you hover your cursor over each advancement, it displays how it is going to be improved.

One important notion concerns the golden, pulsing icon beneath the tab's name (in this case, “Guild”). This same icon appears in every single view in the hamlet overview and it is through that icon that the player can upgrade that particular building to provide their heroes even more benefits. It is very easy to miss/forget and can prevent the player from actually progressing further on down the line as the enemies get more powerful. This is absolutely not obvious to the player, who needs to remind herself constantly of the icon's existence, especially in the beginning.

5. Do I ever need to wait for the interface to load or play an animation?

Perhaps the blockiest usagewise of the three games in this chapter, *Darkest Dungeon* lives and breathes through its animations. Every window and tab has a transition animation which are mostly on par with *Overwatch*'s animations in terms of length. It is a sort of a double-edged sword, while it does provide smoothness and an additional layer of immersion for the player, it also slows down the gameplay. Generally they are tolerable though.

There is one big exception to the rule in the game and that is the score screen after you escape from a dungeon, be it successfully or not. The game lists all the loot you managed to acquire during your crawl through the dungeon. More specifically, it shows all the items in your inventory, one by one, and there is nothing you can do to speed it up or skip it. The player just has to wait there for the game to complete the listing. Especially after adventures that ended poorly this consumes way too much time when all the player wants to do is either quit the game or go fix things in the hamlet. In a game where most other animations and transitions take almost no time at all, a feature like this feels like a slap to the face.

Figure 25: Score/loot screen after a dungeon.

6. Are there any tedious or repetitive tasks that I can shorten (with a shortcut key, for example) or remove entirely?

There are plenty of things that the player does over and over again, either between dungeons or within them. Most of these are reasonable, for example treating diseases needs the players decision-making capabilities as there are only limited slots available each week/turn to do so and multiple heroes may be afflicted with illnesses. Same thing goes for setting up your party or purchasing provisions for a dungeon. Nonetheless, I can't help but wonder if some of them could be more automated, even a little bit. When I'm buying provisions for an adventure, I tend to buy the same things in the same amounts. The only variable usually is the length of the dungeon, but even then I generally have fixed numbers of certain supplies that I take with me. There could definitely be like a recommendation or “save this provision set”-option so the player wouldn't have to click each item separately. Moreover, walking in the dungeons themselves is incredibly repetitive and doesn't really bring much for the player. It could easily be automated so that the party starts moving automatically when clicking on a new room and it stops whenever there is something interesting on the way. Simply giving the player a button to press doesn't mean they want to do so or are playing the game.

Figure 26: The provisioning view. Even though similar items stack, up to a point, the player still has to click each one into their inventory separately.

7. Are the elements of the user interface artistically/graphically in line with the rest of the game? Does anything stand out, and if so, should it?

The overall look of *Darkest Dungeon* is very unique and quite beautiful. Considering that the game was developed by only a handful of people instead of the hundreds that both Blizzard and Bethesda had for *Overwatch* and *Skyrim*, the game is remarkably deep, complex and intricate. Even the menu screens align aesthetically with the rest of the game unlike in the other two games mentioned here. The style of the game doesn't break at any point and provides a constant, highly engulfing atmosphere to the game that is surprisingly rare to find these days.

8. Does the user interface change radically when moving from screen to screen or when opening any of the windows or tabs in it? Are the same elements always in the same place and how do additional information pop-ups fit into the screen?

Differences between all the views and windows in the game can be somewhat big, but keeping to the same style helps mitigate this issue. When the player is in the management-phase, or in between dungeons and watching the hamlet view, all windows and screens that are triggered from the various tabs and buttons on the screen open up inside the hamlet view. They might take over most of the screen, yes, but the hamlet is still in the background even if it's not visible (as there is no transparency to the child-views), so the player stays in the same overall view regardless of what he presses. The content can change even radically, but there is still some weird consistency to the tabs.

Similar actions can be found in the same places, even if the window that is open focuses on something completely different.

Yet again, there is an exception to the majority rule. In this case, it's the view where all the available dungeons are listed. Whenever the player clicks on the red “Embark”-button at the bottom of the screen (shown in figure 21), the view switches from the hamlet to the dungeon selection screen (shown in figure 27). This is a screen of its own and while it shares similar qualities to the hamlet view, it also provides its own information in the dungeon descriptions. In essence, it is yet another tab, but as it is clearly a step into tranferring to the actual gameplay and detached from the actions that can be done in the hamlet, it is justified in having its own screen.

Figure 27: The dungeon selection screen. Party slots are at the bottom of the screen and are filled by dragging heroes into them.

9. How consistent and efficient is the navigation within the interface? Is there anything unneccessary that's only there for show and could become a liability in the long run when players appreciate efficiency over effects?

Considering that most of the game happens either in the hamlet view, in which most of the other screen also open if clicked, or the adventure/combat screen when going through a dungeon, navigation within the interface is generally quite efficient and straightforward. It does have some issues and inconsistencies when looked at more closely though. While most of the time you can close a window or go back to the previous view by pressing a big red X-button on the top corner (as shown in figures 23 and 24), the navigation in both the provisioning screen (figure 26) and the dungeon

selection screen (figure 27) only allow you to back out by clicking the red marks on the top left part of the screen. This is not only very inconsistent with the rest of the game, it is also very unresponsive and unsatisfying to click.

10. Does the player absolutely need all of the items on the screen? Is there a better way to do/show/display something? Should anything else be shown/hidden in each of the views and tabs?

Most of the time, the screen contains mostly useful information and very little trash that could be dumped out. However, after playing the game for a while, there are definitely parts of the UI that the player simply might not need or pay any attention to. This does not mean they should be removed as in some cases they suddenly might become very important for a single decision, but having the screen full of information that you might not even read/use does make it a bit cluttered and begs the question if that space could have been used for something else or even nothing at all.

11. Have you considered people with some sort of disability, for example color blindness? Is your game up to standards in terms of accessibility?

Darkest Dungeon can be played entirely with only a mouse. Every action and selection can be made with the cursor, so that might definitely be easier for some people. While the soundtrack and the sounds of the game are very well done and create a great atmosphere for the game, they are not necessary to play the game and not hearing the sounds doesn't put you at a major disadvantage like for example in *Overwatch*. Visually, though colourful, no single piece of information is represented by only a color and everything has a text decription. Most of the actions, items and skills also have a tooltip if you hover the cursor above them which is a surprisingly important feature in games today and something that isn't a given in some of them.

5. DISCUSSION

The purpose of this chapter is to gather together what has been learned in previous chapters, and even some points that haven't yet been demonstrated, and give a deeper overview of most types of user interface parts and what you should consider in them. It both summarizes the general part of video game UI design and then breaks it down to segments, aiming to provide a clear insight to all of them in turn.

A successful user interface is, at the end of the day, a subjective term. People find immersion and meaningfulness in different things and value them in different ways. There will never be a GUI in a video game that has a 100% acceptance rate among everyone who uses or sees it. All the designer can do is make one that gets as close to it as it can. An integrated user interface isn't an improvement in all cases, sometimes things that clearly shouldn't be there are necessary and actually boost player engagement. In the end, it all comes down to providing the player with information in an efficient way. There's some leeway in this as well, as some people require/desire more guidance than others, but there's a certain range where you can please most of your players. The trick is finding it for every view in your game. This is what it means to design successful video game user interfaces. [21]

If you take only one thing away from this entire thesis, let it be this: your game's user interface is there to enable the player to complete the interactions and actions in the game in the most efficient and satisfying way possible. This doesn't mean that games should not have adversities and not be challenging to the player, it simply means that the challenge needs to come from other sources than terrible controls or uninformative, clunky user interface. A perfect example of this was described by Jerper Juul and Marleigh Norton in 2009:

"Consider the example of a computer-based chess game: Such a game would be badly designed if the interface made it difficult to move the chess pieces. On the other hand, we would expect it to be difficult to decide where to move each piece in order to eventually win the game. This is a clear distinction between the easy interface and the difficult gameplay. We can use usability literature to describe this in further detail: Michel Beaudouin-Lafon distinguishes between the interaction instruments at the disposal of a user, and the domain objects that the user can use the interaction instruments to operate upon. In chess, the interaction instruments should make it easy for the user to move the chess pieces—drag and drop would be an appropriate instrument, by which the player uses the mouse to move the chess pieces directly—but

the chess pieces are domain objects that can be organized in a way that requires the mastery of difficult strategies. Hence we reach a useful distinction between easy interface and difficult gameplay: easy to perform the concrete act of moving the chess pieces according to the rules of the game; difficult to move them strategically well." [15]

The decision of how involved and integrated your user interface is in your game's world is also a subjective concept. Players themselves will generally accept the developer's decision on the matter if it is not blatantly out of place, like an integrated sci-fi-type of HUD in a first person story-driven adventure game like *MYST* (1993) or *Gone Home* (2013). The UI needs to both be and not be a part of the game world. It has the ungrateful task of being an essential control and information tool but also being superimposed on actual game footage and gameplay every time. The trick is finding the right fit and aesthetic appeal so that people see it, accept it, learn it and forget its existence as a separate type of entity while still continuing to use it fluently. [21]

What makes games stand out from other forms of entertainment is their interactivity. In video games this means gameplay, as in the player being able to affect things in the game's fictional world. Video games have no fourth wall as for example movies do and this is one of the reasons accepting an overlay or a fabricated interface is so easy for the players. They need a way to enjoy the experience and interface is that very method. Immersion in games comes from their interactivity as much as any story or realistic graphics. Games give fantasies life as you can choose to live any kind of life in any kind of context and do things you cannot do in reality, like cast magic or slay dragons. The power to bring dreams to life and enable the players to act in them through well-designed user interfaces is the very essence of what makes video games such a unique and incredibly entertaining form of art. [22]

5.1 Lessons learned

Though minimalistic design is a big trend especially in FPS games nowadays, it is not always the way to go. Hiding important and constantly viewed information for the sake of a clearer UI is not an achievement. It all comes down to judging what is important and what isn't and this is not always as clear cut as it may sound. Some people absolutely want an ammo and a health counter for a FPS game, some like immersing themselves without these elements and gauging their health from the visual effects on the screen and counting their shots while firing. Giving players options to hide and display UI elements on the screen provides designers a way to give both types of players what they want, up to a degree. Customization options are a great way to extend your UI to satisfy more players as the UI directly influences the game experience in a major way. [21]

The challenge of the game should come from the gameplay, not the UI. Imagine having to dig a hole in the ground. You'd rather have that the issue was the time and effort you have to spend on it, not the fact that you have to do it without a shovel. Providing players effective tools to do what the game allows them to do is key. When throwing a grenade in a game, you should be more concerned with when, in what angle and how strongly you're throwing it, not how you're going to do it controlswise. [15]

Consistency with other similar games is a surprisingly important factor in games as well. Although you want your game to look unique and be instantly recognizable from its rivals, having similar elements and placing same kind of actions in the same places helps new and old players alike to get accustomed to your interface that much quicker. Players come from a variety of backgrounds and have different experience bases with games when taking up any new game so what might come naturally and seem easy to others might require a lot of learning and teaching for some. This is why a certain consistency even between rivals helps all games to some degree. Then again, the game itself has a lot to do with that as well as even games that belong to the same niche in the same genre can still be worlds apart in many aspects. Nonetheless, stealing and adapting from the best is generally never a bad thing in video game design. [15]

Even though integrated user interface can be a great tool for adding immersion for the player, it is not a self-assured goal to be had in game development. Whenever you're designing a user interface, your primary goals should be ease of use, or intuitivity, and informativeness. Next come consistency and visual appearance, only then should you be concerned whether or not your UI is an integrated one or an imposed one. First decide what needs to be shown, what can be shown and, just as importantly, what shouldn't be shown to the player on your default views and go from there. Do all elements need to be there all the time or should they only appear when necessary or when something has an effect on them? Allow for some customization and accessibility options. All this varies from genre to genre as well, so be sure to check what has worked in your particular genre and in similar games before and even more importantly, what hasn't and why not. [22, 34]

Remember to design your user interface to allow multiple contingency plans, especially from the point of accessibility. If colors are a big part of your interface and colorization is used as a primary game mechanics tool, be sure to use colors that don't limit your playerbase too much and definitely give the players the option to switch the colors. If you use audio cues for effects or important notifications, an underrated tool in modern games, be sure to include some other method of visualizing the same information for maximum player coverage. If possible, allow for multiple different, adjustable control schemes. [34]

While it shouldn't be the single-most crucial factor in UI design, the visual appearance of a game's user interface has a big impact on the players. Bad decisions with it can repel players from the game even if the game itself is solid and enjoyable. Elements on the screen should adhere to an overall aesthetic scheme and be consistent with one another. The styles, images, texts and animations should all look and feel similar so as to build greater user engagement and enjoyment. Same goes for navigation within the interface, make it consistent and logical. Try and keep same actions in same places between screens and make the views as intuitive and understandable as possible so players, both old and new, can understand what is going on in each screen as well as possible without going through a tutorial for it. Sometimes your chosen genre or type of game might not allow it so you shouldn't be afraid of those very same tutorials. Most games have them and they are usually essential. Just make sure you design them to be as informative as possible and don't drag them out for too long, let the player find some things on their own in the actual game itself. Have tooltips in the game itself where possible to smooth out the cognitive load on the player and provide help when needed. [34]

Finally, remember to balance between efficiency and effects. For a new player, all the shine and chrome might make a great impression and provide amazing user experiences for a while, but everything gets old after seeing it repeat over and over again. Eventually the player will become accustomed to playing your game, an expert, and usually wishes to get things done as fast as his skills allow. Your game should not be a limiting factor in this quest for efficiency. If the player has to wait for the interface to play an animation every time they click something, regardless of the length of the animation, it will get extremely annoying. Good examples of this are the activation animation of the Pip-boy in *Fallout 4*, paradigm shifting in *Final Fantasy XIII* or, more familiarly to both the author and the readers of this paper, most animations in *Darkest Dungeon*. One great solution to this is to provide player the chance to skip or disable animations they do not wish to see. Video games hold visual effects in great esteem and this is all fine and well, but do not let it influence your UI design too radically or you risk alienating players in the long run.

5.2 Designing video game UIs

In this section I'll be providing some tips and points for various video game user interface views that appear in many games. Much of what appears in this section is an adaptation of both my own experiences and everything discussed previously in this paper as the intent is to create new views and instructions on video game user interfaces based on them. The focus is more on action games and for example sports games tend to be a bit ignored, but some of these can be applied to them as well. Remember, a user interface is much like a joke: if you have to explain, it wasn't good enough.

One thing I do want to point out at this stage is that designing user interfaces is highly platform-dependent. The exact same interface more often than not does not work on every single platform and hardware. Most common platforms for video games are PC, mobile devices or some form of console like Playstation 4 or Xbox One, which all have their own specifications, formats and capabilities. By default most of the points here relate to PC games as that is perhaps the biggest single platform for bigger games, but many of them can also be applied to games in other platforms with some additional thought. [11]

5.2.1 General UI

The term “general UI” in this context means the main user interface or the view(s) that the player is shown most of the time and contain the core gameplay of the game. This may be the first person view of a shooter where the gun is pointing from the side of the screen and enemies appear deeper in the screen, the hamlet view in *Darkest Dungeon* or the third-person view where you move your avatar in the world in 3D *Final Fantasy* games. It is the main interaction and display method given/shown to the player, which also means it is a crucial part of your interface and requires careful consideration and design to succeed in its task, both short- and long-term.

The trend in many games seems to be favouring a minimalistic design where the user interface is stripped of most of its static elements. They are displayed only when necessary, for example a health bar is shown when the player is taking damage. Another approach to this is to dump the health bar entirely and display the player's health status as a visual cue as shown in figure 12. Most favourable option most likely stands somewhere in between. While I'm an advocate for immersion and not cramming everything possible into a single user interface view, not displaying information just for the sake of a more clear UI is not a smart option either. Visual cues to different changes on the screen are great indications that something is happening and the player needs to be aware of it. Another option is what is called “hover controls”, where elements on the screen are hidden or collapsed as a default and pop-up or open up when the player brings their controller over them. *Destiny* (2014) embraces this philosophy and its UI is widely regarded as a great success on both PC and consoles, which is quite a feat to

accomplish. [7, 8, 12, 38]

The easy solution to whether or not you should be showing a certain element on the screen is to not make the decision yourself at all. Let the players themselves decide on what they want to see and what they don't and whether elements are staticly displayed or only when they change. Customization is the name of the game, ask any *World of Warcraft* player or even a developer. While you as a designer do need to give the player a blueprint and a default view for the game which is in use for at least the tutorial's duration, in many cases you can then tell the player that they can go and customize settings from the options if they so choose. Do not leave this up to the modding community (in many games you can't as the game is not modifiable) but take the initiative and let the players decide what they want if at all possible.

Sometimes, however, you absolutely have to have a fixed set of elements on the screen to be able to provide the player all the necessary information about a given situation. Usually these games are heavily action/reaction-based, like FPS or MOBA games, which also carry the weight of not being modifiable to prevent cheating. Console and mobile games also almost never include customization options as the platforms themselves are more limited in that regard. Designers then can go back to the fundamental questions, like what exactly do we want to enable the player to do, what is the aim of the game and what information is directly related to the actions the players take in the game. After that, rigorous testing and iteration is required. You do not want to become notorious for releasing games or updates that seem like nobody tried them out before asking money for them. The video game industry has more than enough of these failure stories already. Great way to get started is to look back on other games and figure out what they did right or wrong, for example similarly as in chapter 4 of this thesis. Don't get too bogged down with only copying what works though, innovation always has a place in this industry and is one of the driving forces of great success stories. Just go ask Markus Persson, the creator of *Minecraft* (2011).

At the end of the day, you need to find the right fit for your idea. It is not too different from creating many other products. You have a goal, to entertain the players or to tell a story or simply get your idea into a more concrete form, and you have tools to make it happen. Nobody can tell you what you can do with your idea, even though some actions and decisions are strongly advised or discouraged as seen in chapter 4 as well. Simple belief in your idea will get you far even if it doesn't guarantee anyone else will like it. That's where instructions like this come in and help you form it in a more approachable manner.

5.2.2 Tutorial

Tutorial interfaces deserve their own sub-section for two important reasons: first, they are very common and they are usually the very first thing the player experiences in a new game that teaches them how everything works and how to accomplish things

within the game. Second, while tutorials are usually built on top of the actual game and many times masqueraded as a sort of an introduction to the actual story, they also have additional interface elements or layers for visual cues on how to do the current goal of the tutorial part. Most of the time these instructions are made in similar fashion as buttons notifications or reminders later on in the game, but regardless, it brings additional information to the screen that usually is not there.

Tutorials have the added benefit of being able to introduce the entire UI to the player in practically any manner they want without breaking the players flow or bothering them overmuch. As shown in figure 28, the tutorial information is clearly way out of place in the actual game itself, yet as this is literally from the very first gameplay minute and the player does not yet know anything about how to survive in the game, the text is more than welcome as it provides instructions on how everything works. Visually, the style needs to be in line with the rest of the game for it to be a tool for immersion. As the figure below shows, it doesn't take much for aesthetic unity.

Figure 28: Tutorial in *Lords of the Fallen* (2014)

Another incredible tool for tutorials is progressive disclosure. What this means is that despite a game having complex controls and many aspects that the player has to go through in the tutorial to get the basic hang of, say, commanding every kind of unit in a strategy game like *Tom Clancy's Endwar* (2008), everything is not simply thrown at the player all at once but divided into various smaller parts and tutorial segments. These parts are then introduced to the player one after another so they can get an understanding of each different type of aspect that the game has. The player also then has the option of skipping some parts or pausing and continuing at a later time. Tutorials don't necessarily always end at the start but the game can introduce new aspects even further on down the line when they become relevant. This help smooth out the cognitive

load on the player and it also extends the game's lifespan as new features are introduced throughout the game, not only at the beginning. [12]

A great way to do a very thorough tutorial, regardless of the genre of your game, is to go through each element on your screen one by one and explain, or even show with gameplay videos, what they all do. Naturally you don't necessarily need to do this with every single button and icon (for example, a red X button in the upper corner of a pop-up window very rarely means anything other than “close this”), but for all the custom buttons and important gameplay elements you should do this. Provide the player the option to check the elements again from a menu or somewhere if they wish to, don't assume they remember everything on the first try. Of course, if at all possible, provide customization options.

5.2.3 Information view (inventory, character details etc.)

More often than not, some sort of information view(s) are the beating heart of a game and especially any kind of management you do in it. Regardless of whether you need to organize your backpack to find that double-handed greatsword of lifestealing in a RPG, check your item build in *Dota 2* or try to figure out which building type would best suit that particular stretch of land in a city-building simulator, information is necessary and crucial. You also generally require a lot of it, which makes the way it is displayed to the player immensely important. Games have the additional problem that their way of displaying information not only needs to be efficient, it also needs to look good. No excel-sheets for game designers, sorry.

Unlike in most other views (quantitywise), information is king in these types of windows. You don't generally have to be scared of filling up the screen with information as that is the very point of inventories, character sheets or any and all kinds of lists that games, and other software and products as well, use for just such a case. Naturally you need to be somewhat selective of what data you show, as not everything about a certain item for example are things that the players generally look at. In a RPG, you for example would always want to display the name and type of a weapon and how much damage it does along with any specifications that relate to its function or effects. A text description, or the lore of the weapon, is a good thing to have in the game, but it is something that most players do not need to see most of the time and usually don't mind looking up from behind a separate tab or button if it doesn't fit into the same screen smoothly.

You also need to provide layering for the player. Not everything fits into a single screen so take advantage of paging, tabs and sorting your information views with hotkeys and quickbuttons. Show the player the top level of information for quick reference and offer them the possibility to look further into details on anything

particular if they so choose, but don't force them to do so. For example in *Shogun II: Total War*, displayed in figure 11, your units are arranged by their type or set group and are represented by icons on the bottom bar. Most of the time you only need to view the current situation of the unit, is it fighting or able to do so and double-click on it to move the camera on top of it. Further details are always available by right-clicking on the unit icon, but most of the time you do not need them to utilize the unit in an efficient way.

Continuing with the same example, another helpful method of displaying information is hover controls as mentioned earlier in this section. Much of the information can be hidden until when you move the cursor on top of a certain location, in this example a unit icon, when another window pops up and displays the statistics of that unit as numbers. This is a clever way to hide more information in plain sight so that it is very easily and quickly reachable without cluttering the screen all the time.

More often than not, information screens pause the game or at least happen during times when there aren't too much action going on in the background (which in and of itself is a good design decision), so the player can comfortably browse through the large amount of information for whatever they need. This means that the entire screen is your playground. Just remember to stay visually within the colour palette and style of the game so the player doesn't get confused when the window shows up to take up the entire screen. Also be clear with your navigation possibilities, what does what and display hotkeys if at all possible. Try to keep controls similar, for example having the escape-button on the keyboard always close the current information tab or go back to the previous one so that if the player gets lost, they can always back out and regroup.

5.2.4 Maps & navigation

Rarely does a game center around one single, miniscule location (like an arena or a single room of a house) or a single view, which contains everything that happens when the actual gameplay happens. Nearly all games therefore need to think about how the player navigates in the game world. It can be the actual moving of the avatar or simply repositioning the camera, or even just switching from a menu screen to another, but navigation in all its complex simplicity is a crucial and rarely spoken about factor in many games. The same goes for any kind of map or minimap which becomes essential every time the player cannot see everything by simply spinning the camera around. Which is, again, nearly always.

Navigation can mean anything from a breadcrumb trail through the game's menus to the actual gameplay and you looking for the direction of your next objective from an in-game minimap or compass. Regardless of what kind of navigation it is, the most important aspect of it is clarity. There is no more important function for navigation than to provide clear routes and instructions on how to proceed or backtrack your steps, both in menus and in gameplay. If the navigation is not made clear to the player, they may get lost or stuck which immediately both breaks the player's flow and immersion and

deals a serious blow to their enthusiasm to continue playing the game.

Menu navigation is a straightforward thing, or at least it should be. Menus are mostly static and you do not need to switch around the positioning of mostly used buttons, in which category menu buttons definitely belong. Position them in similar places and adapt a unified visual style so the player always knows where to look for navigation elements and what they look like. Again, provide textual/visual instructions or hotkey icons for the player as clicking a hotkey on a keyboard or controller is generally faster than moving the cursor around on the screen. *Overwatch* does this and it is usually very efficient and satisfying to browse through the game's menus.

Map views have a few more variables, but ultimately are not very complex either. Perhaps somewhat unexpectedly, minimaps might be the most difficult designwise to implement efficiently as the space you have on the interface is limited yet you still need to cram most of the map's information in it in some format. Maps themselves are just another form of an information screen, meaning you can take over the entire view without a worry and focus on making the content as informative and goal-oriented as possible.

One thing to remember both on minimaps and actual map views is that they are rarely static nor should they be. Bringing life to a flat, boring picture of a map via animations, icons, colors and interactable elements is great map design. You need to allow the player to be able to click on various points of the map to see more information about them, move the map and turn it around freely and usually set markers and waypoints for themselves. Zooming in and out of the map helps distinguish between levels of details even though it adds another layer of designing for the developers.

5.2.5 Combat

More often than not action-based and story-driven games involve some form of conflict in them. This requires the players to take actions within the boundaries of the game to solve the problem with the tools they have available. Or, to put it into layman's terms, hit the other guy on the head before he does so to you in the game. Combat is one of the main lures of many of the biggest games on the market nowadays, be it a FPS game like *Battlefield 1* (2016) or a massive multiplayer RPG like *World of Warcraft*. Sooner or later, there will be violence and you need to be well equipped to handle it from the user interface perspective as well.

How you handle combat depends largely on the genre in question as there aren't any over-arching solutions that fit every situation. In FPS games success usually comes from the player's own reactions as controls have a bigger role than user interface in the long run (if it's “good enough”). Strategy games rely a bit more on the user interface, although hotkeys and decision-making from the player are the keys to victory. RPGs tend to be a bit slower paced and they rely pretty heavily on usable interfaces for great

player experiences. Massive multiplayer online games are perhaps one of the most important genres user interface-wise as though they can be more often than not customized to fit your specific needs, it is still a UI design decision to allow it and you still need to provide a default interface and think about all the elements that the interface needs.

As combat can take such a variety of forms, user interface design should, as a general principle, revert back to the very basic principle of enabling the player to accomplish the core idea of the game and work its way up from there towards their particular niche. If you have a game where you control a party of adventurers in turn-based combat, your focus needs to be on displaying the necessary information about each character, their abilities and how the player can use them in the way they see fit as well as move them around on the battlefield to gain positional advantage.

More importantly, combat is a very focused area of any game. Equally, if not even more so, important is what you leave out of that view as too much information can quickly become burdensome. Combat is meant to be fast-paced and characters lives are at stake, you need to give your full attention to it or bad things might happen. Combat is about efficiency, about predicting your opponent and calculating the best chances of successfully winning that particular scuffle. The user interface needs to reflect this and provide tools and information to support this goal.

Many FPS games put a lot of information on the screen, as can be seen with examples from *Overwatch*, and it can work as most of the information is something the player can use when making his decisions. Then again, some games forgo displaying everything alltogether, like *Skyrim* or *Call of Duty: Modern Warfare* on hardcore-mode, and might achieve just as good of a result. This simply goes to show that you can do many things in regards to your UI design and they could work, it is simply a matter of finding the right fit. The best advice here is to seek examples from previous games of the same genre and analyze for yourself why something works and why something doesn't and how well something might suit your game. The faster the pace of the game, the more important the controls and displayed information are. Minimalism is not a goal in itself, usability is. Support the player's decision-making processes and provide tools for it while also clearly displaying the game's reactions to their actions.

5.2.6 Cutscenes

While cutscenes and cinematic story-telling is in most cases less time-consuming than actual gameplay, it is the single-most important aspect of creating immersive stories and building deep player experiences. Every game has a context to it which can be ultimately boiled down to telling why the player is doing what she is tasked with doing. Some games just take more time to build it up and emphazise it more. Story-telling is many times done with cut-scenes, or clips of animation where the player has no or very minimal control. To the overall game experience, these scenes are essential.

Story scenes differ more radically from the rest of the game than any other part of it in terms of user interface. They are almost always there to progress the story and provide depth for the characters, reasons why things are happening and build emotional attachment for the player. Usually, there is no interaction to be had in these screens and any interface elements related to gameplay would actually hamper the player experience. There are examples to the rule, sure, but mostly cutscenes are miniature animation movies where the player is simply an observer. This makes it easy for the designer as they simply need to remove the ability to click on anything, even by accident so the player recognizes “Oh, storytime, I can focus on what's happening instead of reacting to it immediately”.

The issue with cutscenes is the tranfer from and to them. Depending on the genre and the game in question, sometimes the moments leading up to a cutscene are clearly marked, such as a savepoint before a door or the start of a mission. Generally there is some sort of tip in the game so the player understands that a videoclip will probably play once they do something. Sometimes, however, this is not the case and the player might get surprised by the camera angle suddenly changing or the graphics quality taking a cinematic style as a pre-recorded clip starts playing. It is important to prepare for the player accidentally clicking something or pressing a button. Some games, especially older games, fail to do this and as a result a surprising cutscene might get skipped altogether. Offer the player the chance to skip cutscenes, unskippable ones are horrible design decisions, but do it so that they know what they are doing. Take for example the following cutscene from *Lords of the Fallen*:

Figure 29: Cutscene from *Lords of the Fallen* with the possibility to skip it.

The clip plays with no prompts or hotkeys if the player does not press anything. Most

PC games offer the chance to skip cutscenes with the escape-button and if the player presses that, it displays the shown message in the top-right corner. This serves as a safeguard for accidental presses while offering the player the chance to jump into the game faster if they so choose. It is a safe way to provide player control over cutscenes.

Cutscenes can be played in the regular gameplay view as well, but in that case you should disable the regular UI view and buttons that the player normally uses so they won't misinterpret the scene as interactable. Most developers do not practice this method as storytelling is more natural in their own scenes and views, but it can be done if you absolutely want to or if the cutscene is really short. Whether you should or not depends, again, entirely on your particular situation.

6. SUMMARY & CONCLUSION

Video games have come a long way since their inception and they have shown incredible transformation and progression capability in a comparatively short time period as can be seen from the history recap of this thesis. The pace in which video games have become a dominant force in entertainment is practically unheard of and it shows no signs of slowing down, especially with the recent introductions to consumer-grade virtual and augmented reality concepts and devices. They were omitted from this thesis as they are thesis-worthy topics in their own right and while augmented reality has much in common with ”traditional” video game user interface design, virtual reality in particular brings about a whole new set of challenges and development issues.

During my 20 years of gaming I've come across quite a lot of video game user interfaces on a great variety of platforms, from the Nintendo Entertainment System, or NES, to modern PCs and consoles. The fact that it is still a mere fraction of all those designed and published speaks volumes on the scale of the industry. Many games, no matter how good they are, are doomed to oblivion as the sheer mass of released games every week swallows even diamonds amidst the trash.

This is another factor why user interface design is absolutely crucial. As can be seen from this thesis as well, creating a great video game user interface is not simply ”copy the UI elements from that game because it worked”. It requires incredible amounts of planning, designing and iterating, but most of all, endless amounts of testing. The greatest of user interfaces can be incredibly minimalistic and show next to nothing, like in *Skyrim*, but this doesn't mean that a lot of people didn't spend countless of hours iterating it to its final form. User interface design is a lot more complex process than some might give it credit for and its importance cannot be understated as the user interface views are the very things that the player is looking at most of the time and how they interact with the game. If they do not work or are arduous to use, it can easily alienate players and that can easily start a snowball towards bankcrupty for the developing company as especially smaller ones are usually highly reliant on the success of each of their released games.

As I said in the empirical part of the thesis, if there is only one thing you take away from reading this thesis, it is this: your game's user interface is there to enable the player to complete the interactions and actions in the game in the most efficient and satisfying way possible. The thing is, ”the most efficient method” for the player generally still requires some actions or effort within the game context. If you're playing

a FPS, your goal is usually to eliminate the opposition. The most efficient way of doing this would be to press a button and they would all drop, but the game has rules on how you can do this. You still need to find them, not get shot yourself and use precice hand movements to aim and timing to click the mouse to fire the gun. Games are about creating desirable challenge for the player and allowing them to overcome those challenges within the boundaries of the game which is then rewarded somehow, be it by progressing the story or giving them something like a new accessory in the game.

Different game genres have different focus points and different prioritization areas, but the same principles still apply for user interface design on a basic level. It's when you get to more specific details on higher levels of the user interface that the genres start to distinguish themselves, but deep down they all need to ask themselves the same questions. What do I need to show to the player? Does it need to be always visible? What absolutely doesn't need to be there? How do I position everything to be as logical as possible? How does navigation work in the game? Are my graphical aestethics unified? The list goes on and can be found from chapter 4 of this thesis as well. Answering those questions gets you a long way with your user interface design even if you have no idea what you're doing.

Going back to the goal of this thesis, answering the questions of how can I design a great user interface and what do I need to consider when designing it, I feel the paper has provided helpful ways to do this (chapter 4) and created novelty value by forming guidelines for video game user interface design in specific as that is one area I did not find much previous work in. User interface design is indeed a well-researched expertise area, but transferring that expertise to video games and taking into account all the different unique factors of video games is something that hasn't been done too much previously.

If this thesis was to be further expanded, the next logical step would be to create user tests on a variety of games, for example the three analysed in detail in chapter 4, and have actual players, both new and experienced ones, give their thoughts about the interfaces. They could first be freely interviewed and then the list of design questions provided in this thesis could be reviewed with them so as to gain specific knowledge about each particular area and validify and add/modify the questions themselves.

REFERENCES

[1] Staff Writer, The biggest entertainment markets in the world, May 2015, http://businesstech.co.za/news/lifestyle/88472/the-biggest-entertainment-markets-in-the-world/

[2] Laura Ermi, Frans Mäyrä, Fundamental Components of the Gameplay Experience: Analysing Immersion, Proceedings of DiGRA 2005 Conference, Changing Views – Worlds in Play, 2005, http://www.digra.org/wp-content/uploads/digital-library/06276.41516.pdf

[3] Tom Senior, The 13 worst game design crimes, January 2013, http://www.pcgamer.com/game-design-sins/1/

[4] Jez Corden, Top 5 worst video game design mistakes, June 2015, http://www.windowscentral.com/top-5-worst-game-design-mistakes#slide1

[5] Ernest Adams, The Designer's Notebook: Bad Game Designer, No Twinkie! IX, October 2008, http://www.gamasutra.com/view/feature/3812/the_designers_notebook_bad_game_.php?print=1

[6] Ian Fisch, 10 Game Design Process Pitfalls, May 2009, http://www.gamasutra.com/view/feature/4017/10_game_design_process_pitfalls.php?print=1

[7] James Plafke, Bungie did something incredible with *Destiny*'s user interface, September 2014, http://www.geek.com/games/bungie-did-something-incredible-with-destinys-user-interface-1604841/

[8] Mike Bithell, What to steal: From *Destiny*'s UI, 2015, http://mikebithell.tumblr.com/post/97214168242/what-to-steal-from-destinys-ui

[9] Desi Quintans, Game UI By Example: A Crash Course in the Good and the Bad, January 2013, http://gamedevelopment.tutsplus.com/tutorials/game-ui-by-example-a-crash-course-in-the-good-and-the-bad--gamedev-3943

[10] Sean Targett, Victoria Verlysdonk, Howard J. Hamilton, Daryl Hepting, A Study of User Interface Modifications in World of Warcraft, Game Studies, Vol. 12, Issue 2, December 2012, http://gamestudies.org/1202/articles/ui_mod_in_wow

[11] Jakob Nielsen, The Difference Between Web Design and GUI Design, Nielsen Norman Group, May 1997, https://www.nngroup.com/articles/the-difference-between-web-design-and-gui-design/

[12] Janko Jovanovic, 10 UI Design Patterns You Should Be Paying Attention To, June 2009, https://www.smashingmagazine.com/2009/06/10-ui-design-patterns-you-should-be-paying-attention-to/

[13] Nadi Awad, Game Development as Software Development, March 2014, https://blog.inf.ed.ac.uk/sapm/2014/03/13/game-development-as-software-development/

[14] Laura Ermi, Frans Mäyrä, Player-Centred Game Design: Experiences in Using Scenario Study to Inform Mobile Game Design, Game Studies, Vol. 5, Issue 1, October 2005, http://www.gamestudies.org/0501/ermi_mayra/

[15] Jesper Juul, Marleigh Norton, Easy to Use and Incredibly Difficult: On the Mythical Border between Interface and Gameplay, Proceedings of the 4th International Conference on the Foundations of Digital Games, ACM (2009), pages 107-112., April 2009, https://www.jesperjuul.net/text/easydifficult/

[16] Gonzalo Frasca, Simulation versus Narrative: Introduction to Ludology, Routledge, 2003, http://www.ludology.org/articles/VGT_final.pdf

[17] Ryan Taljonick, Why Final Fantasy 14 failed (and how it was reborn), March 2014, http://www.gamesradar.com/why-final-fantasy-14-failed-and-how-it-was-reborn/

[18] Kai Huotari, Juho Hamari, Defining Gamification - A Service Marketing Perspective, MindTrek '12 Proceeding of the International Academic MindTrek Conference, pages 17-22, October 2012, http://dl.acm.org/citation.cfm?id=2393137

[19] Amanda Kudler, Timeline: Video Games, Infoplease, http://www.infoplease.com/spot/gamestimeline1.html

[20] Video game history timeline, http://www.museumofplay.org/about/icheg/video-game-history/timeline

[21] Stein C. Llanos, Kristine Jørgensen, Do Players Prefer Integrated User Interfaces? A Qualitative Study of Game UI Design Issues, Proceedings of DiGRA 2011 Conference: Think Design Play, 2015, http://www.digra.org/wp-content/uploads/digital-library/11313.34398.pdf

[22] Kristine Jørgensen, The User Interface Continuum: A Study Of Player Preference, April 2011, http://www.gamasutra.com/view/feature/6346/the_user_interface_continuum_a_.php

[23] Casual game definition, 2016, https://en.wikipedia.org/wiki/Casual_game

[24] Simulation game definition, 2016, https://en.wikipedia.org/wiki/Simulation_video_game

[25] Valentijn Visch, Niko Vegt, Hester Anderiesen, Katinka van der Kooij, Persuasive game design: A model and its definitions, May 2013, http://gamification-research.org/wp-content/uploads/2013/03/Visch_etal.pdf

[26] Anthony Stonehouse, User interface design in video games, February 2014, http://gamasutra.com/blogs/AnthonyStonehouse/20140227/211823/User_interface_design_in_video_games.php

[27] Erik Fagerholt, Magnus Lorentzon, Beyond the HUD - User Interfaces for Increased Player Immersion in FPS Games, Göteborg, Chalmers University of Technology, 2009, http://studentarbeten.chalmers.se/publication/111921-beyond-the-hud-user-interfaces-for-increased-player-immersion-in-fps-games

[28] Om Tandon, UX Deep Dive: Painful reality of building FPS experiences on mobile? - Part 1, June 2016, http://www.gamasutra.com/blogs/OmTandon/20160613/274776/UX_Deep_Dive_Painful_reality_of_building_FPS_experiences_on_mobile__Part_1.php

[29] Sebastian Deterding, Dan Dixon, Rilla Khaled, Lennart Nacke, From Game Design Elements to Gamefulness: Defining "Gamification", MindTrek '11 Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, pages 9-15, September 2015, http://dl.acm.org/citation.cfm?id=2181040

[30] Kevin Werbach, (Re)Defining Gamification: A Process Approach, Persuasive Technology: 9th International Conference, PERSUASIVE 2014, pages 266-272, May 2014, http://link.springer.com/chapter/10.1007/978-3-319-07127-5_23

[31] Stephen Dale, Gamification: Making work fun, or making fun of work?, January 2015, http://www.slideshare.net/stephendale/gamification-making-work-fun

[32] Laura Dabbish, Luis von Ahn, Designing Games With A Purpose, Communications of the ACM, Vol. 51, Issue 8, pages 58-67, August 2008, http://dl.acm.org/citation.cfm?id=1378719

[33] Joel Couture, Videoball and the challenge of designing (and selling) simplicity, July 2016, http://www.gamasutra.com/view/news/277949/Videoball_and_the_challenge_of_designing_and_selling_simplicity.php

[34] Greg Wilson, Off With Their HUDs!: Rethinking the Heads-Up Display in Console Game Design, February 2006, http://www.gamasutra.com/view/feature/2538/off_with_their_huds_rethinking_.php

[35] Game accessibility guidelines, 2012-2015, http://gameaccessibilityguidelines.com/full-list

[36] Jakob Nielsen, 10 Usability Heuristics for User Interface Design, Nielsen Norman Group, January 1995, https://www.nngroup.com/articles/ten-usability-heuristics/

[37] Heather Desurvire, Charlotte Wiberg, Game Usability Heuristics (PLAY) For Evaluating and Designing Better Games: The Next Iteration, OCSC '09 Proceedings of the 3rd International Conference on Online Communities and Social Computing: Held as part of HCI International 2009, pages 557-566, July 2009, http://ocw.metu.edu.tr/pluginfile.php/4129/mod_resource/content/0/ceit706_2/10/game_usability-_heuristics.pdf

[38] Bryant Francis, Building the user interface for Bungie's *Destiny*, June 2016, http://www.gamasutra.com/view/news/275254/Video_Building_the_user_interface_for_Bungies_Destiny.php

RELATED READING MATERIAL

Key user experience insights gained during the creation of *Dreadnought*'s combat HUD
http://www.gamasutra.com/blogs/OliverJanoschek/20160714/277048/Key_user_experience_insights_gained_during_the_creation_of_Dreadnoughts_combat_HUD.php

Invisible Walls
http://www.gamecareerguide.com/features/593/invisible_.php

The afterimage of *Final Fantasy VII*
http://www.gamasutra.com/view/news/263174/The_afterimage_of_Final_Fantasy_VII.php

Take the time to appreciate a game's buttons and menus
http://kotaku.com/take-the-time-to-appreciate-a-games-buttons-and-menus-1592990119

ReCore developers had to consider accessibility for its color-driven gameplay
http://www.gamasutra.com/view/news/275392/ReCore_developers_had_to_consider_accessibility_for_its_colordriven_gameplay.php

How Bad User Interfaces Are Ruining Video Games And Other Wonderful Things
http://www.kotaku.com.au/2012/05/how-bad-user-interfaces-are-ruining-video-games-and-other-wonderful-things/

Game UX Summit 2016
http://www.gamasutra.com/blogs/CeliaHodent/20160722/277651/Game_UX_Summit_2016__All_Sessions_Summary.php